

# **FNDE**

*Fundo Nacional  
de Desenvolvimento  
da Educação*

## **Guia de Contagem do FNDE**

## 1. Sumário

<b>2.</b>	<b>INTRODUÇÃO .....</b>	<b>4</b>
2.1.	Base Instalada de Aplicações do FNDE .....	4
<b>3.</b>	<b>PRINCÍPIOS NECESSÁRIOS À ANÁLISE DE PONTOS DE FUNÇÃO .....</b>	<b>5</b>
3.1.	Orientações para Distinguir Requisitos de Usabilidade.....	5
3.2.	Orientações para Distinguir Requisitos de Confiabilidade .....	6
3.3.	Orientações para Distinguir Requisitos de Desempenho.....	7
3.4.	Orientação Geral .....	7
<b>4.</b>	<b>FATOR DE AJUSTE .....</b>	<b>8</b>
<b>5.</b>	<b>DEFLATOR CONFORME O TIPO DE AÇÃO NA FUNÇÃO .....</b>	<b>9</b>
<b>6.</b>	<b>MEDIÇÃO DAS MANUTENÇÕES.....</b>	<b>10</b>
6.1.	Dicas para otimizar o custo das manutenções .....	10
6.2.	Campos Não Editáveis ( <i>Read-Only</i> ) .....	12
6.2.1.	Cenário I: Campo preenchido automaticamente para restringir valores de entrada do usuário, de acordo com o perfil do usuário autenticado. ....	12
6.2.2.	Cenário II: Campos preenchidos automaticamente em uma consulta implícita para alteração de dados .....	13
6.2.3.	Cenário III: Campos de cabeçalho apresentados ao longo de todo o sistema .....	13
6.3.	Subdivisão de uma Funcionalidade em <i>n</i> Funções.....	14
6.3.1.	Cenário I: Mais de um <i>Processo Elementar</i> identificado para a funcionalidade .....	14
6.3.2.	Cenário II: Um único <i>Processo Elementar</i> identificado para a funcionalidade.....	15
6.4.	Impressão de Termos e Formulários .....	17
6.4.1.	Cenário I: Formulário composto por dados resgatados de vários <i>Arquivos Lógicos</i> .....	17
6.5.	Envio de Email.....	18
6.5.1.	Cenário I: Envio de e-mail como consequência de um processo .....	18
6.5.2.	Cenário II: Envio apenas do e-mail, independente de outras transações .....	19
6.6.	Identificação de Tipos de Registro e Arquivos Lógicos .....	19
6.7.	Conversão de Dados.....	22
6.7.1.	Cenário I: Migração de dados dissociada de projeto.....	23
6.7.2.	Cenário II: Migrações de dados frequentes/periódicas.....	23
6.8.	Middleware.....	24
6.8.1.	Cenário I: Sistema intermediário.....	24
6.8.2.	Cenário II: WebService dentro da fronteira da aplicação .....	26
6.9.	Processos Sem Interface Gráfica (batch) .....	27
6.9.1.	Cenário I: Rotina Batch Não-Mensurável .....	28
6.9.2.	Cenário II: Rotina Batch Mensurável.....	28
6.9.3.	Cenário III: Rotina Batch como parte de outros Processos.....	29

6.10.	Data Warehouse e BI.....	29
6.10.1.	Cenário I: Exemplo de <i>Data Warehouse</i> .....	30
6.11.	Workflow.....	33
6.11.1.	Cenário I: Exemplo de Workflow .....	34
6.12.	Views/Visões em Bancos de Dados .....	35
6.12.1.	Cenário I: Views/Visões úteis durante a identificação de Arquivos Lógicos.....	35
6.12.2.	Cenário II: Views/Visões não úteis durante a identificação de Arquivos Lógicos....	36
6.13.	Transações compartilhadas entre várias aplicações.....	36
6.14.	Consultas com múltiplas mídias .....	37
6.14.1.	Cenário I: Relatório com múltiplas mídias.....	38
6.15.	Funções transacionais acessadas por diversos perfis de usuário .....	39
6.16.	Consultas/Pesquisas com diversas opções de filtro .....	40
6.16.1.	Cenário I: Consulta com um único Processo Elementar .....	40
6.16.2.	Cenário II: Consulta com vários Processos Elementares .....	41
6.17.	Trilha de Auditoria.....	42
<b>7.</b>	<b>ITENS NÃO MENSURÁVEIS (INM) PELA APF.....</b>	<b>44</b>
7.1.	Manutenção Corretiva .....	44
7.2.	Manutenção Cosmética .....	46
7.3.	Manutenção Técnica (ou Adaptativa) .....	47
7.4.	Dados de código .....	48
7.5.	Tabelas de Itens Não Mensuráveis.....	48
7.5.1.	INM associados a funcionalidades.....	49
7.5.2.	INM associados a características gerais da aplicação.....	50
<b>8.</b>	<b>ESCOPO DO SERVIÇO EM PF .....</b>	<b>51</b>
8.1.	Atividades Inclusas no Preço do PF .....	51
8.2.	Atividades NÃO Inclusas no Preço do PF.....	51
<b>9.</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>52</b>

## 2. Introdução

---

Este documento tem a finalidade de facilitar o uso da técnica de pontos de função dentro do contexto de desenvolvimento e manutenção de sistemas do FNDE. Este guia funciona como um **complemento** ao Manual de Práticas de Contagem do IFPUG (*International Function Point Users Group* – [www.ifpug.org](http://www.ifpug.org)), versão 4.3, que estabelece todo o padrão para a contagem de pontos de função.

A idéia é que o guia torne mais prático o uso dos conceitos e regras definidos pelo IFPUG com a sua exemplificação em situações peculiares ao FNDE e para situações em que o IFPUG não oferece orientação prática ou objetiva, este guia irá suprir estas deficiências.

### 2.1. Base Instalada de Aplicações do FNDE

---

Vide documentos disponíveis no diretório: “..\Núcleo de Métricas\Base Instalada\” da ferramenta StarTeam®.

### **3. Princípios Necessários à Análise de Pontos de Função**

---

Os insumos para Análise de Pontos de Função (APF) são os requisitos funcionais do usuário. Existe uma grande convergência sobre o que seja requisito funcional e quais os artefatos gerados no processo de desenvolvimento que os capturam em teoria; contudo, verifica-se no mercado brasileiro na data da publicação deste documento haver pouco rigor ou pouca preocupação em evidenciar de maneira clara e consistente essa condição ao analisar os resultados da elicitação dos requisitos dos usuários na produção dos documentos de especificação de requisitos (como casos de uso, documentos de visão, etc.) para fins de desenvolvimento e manutenção de sistemas.

Para fins de medição e estimativa do tamanho funcional, requisitos funcionais são aqueles particulares e específicos de uma tarefa ou serviço do usuário; representados pela capacidade do produto atender às suas necessidades de interação entre o sistema ou de armazenar seus dados. Esse processo deve desconsiderar aspectos que correspondem às restrições de ordem geral como:

- Restrições de qualidade como definidas pela (confiabilidade, usabilidade, eficiência, facilidade de manutenção e portabilidade);
- Restrições relacionadas às tecnologias utilizadas na implementação (sistemas operacionais, linguagens de programação, sistemas gerenciadores de banco de dados, interfaces gráficas com o usuário, ferramentas de testes, etc.);
- Restrições do ambiente (interoperabilidade e proteção de danos acidentais ou intencionais); ou
- Restrições quanto à organização (diferentes locais para operação, equipamento alvo, aderência a padrões).

Os itens relacionados nesta lista tem o propósito de ilustrar diferentes tipos de restrições gerais, não de esgotar todos os tipos possíveis. Visando enriquecer este documento e ilustrar o que se deseja dizer por de maneira clara e consistente, é apresentado um exemplo na distinção de um requisito de usabilidade.

#### **3.1. Orientações para Distinguir Requisitos de Usabilidade**

---

Ao analisar um conjunto de casos de uso, interfaces com o usuário, relatórios e interfaces com outros sistemas que tenham indícios de representarem partes de um processo de negócio maior e incompleto se apenas cada parte for realizada isoladamente como, por exemplo, telas estruturadas em abas, sequência de telas como em "wizards", quadros em um formulário maior, telas separadas de dados funcionalmente dependentes entre si, etc., deve-se fazer a seguinte pergunta ao especialista no assunto:

“Fossem esses passos realizados todos de uma vez, por um mesmo usuário desempenhando um mesmo papel, a necessidade de negócio está atendida independentemente da maior dificuldade de entrar os dados ou mesmo da aparência estética da tela ou formulário?”.

Haver uma infraestrutura que provê a possibilidade de controlar acesso até ao nível de campo, dificulta essa análise na medida em que todo pequeno passo que compõe um processo (em potencial) pode ser alocado a um perfil em particular; essa infraestrutura geral não deve ser considerada na análise. O que deve ser considerado são os requisitos particulares e específicos de uma transação.

### **3.2. Orientações para Distinguir Requisitos de Confiabilidade**

---

De maneira similar à orientação anterior quando avaliando um conjunto de casos de uso e/ou interfaces com o usuário e haja indícios de representarem partes de um processo de negócio maior e incompleto se apenas cada parte for realizada isoladamente como, por exemplo, telas estruturadas em abas, sequência de telas como em “wizards”, quadros em um formulário maior, telas separadas de dados funcionalmente dependentes entre si, etc. e onde em cada passo do processo os dados sejam salvos, faça a seguinte pergunta ao especialista no assunto:

“O motivo pelo qual esses dados são salvos em cada etapa é relativo à instabilidade do ambiente? Por exemplo, em sistemas Web é comum ao cadastrar um pedido, a cada produto incluído no pedido, cada alteração nos detalhes do pedido de um produto ou quando se exclui um produto do pedido, haver a salva da operação para evitar uma nova digitação. Não fosse essa instabilidade, haveria a preparação de todo o pedido e apenas ao final da edição desse pedido, o mesmo seria salvo como um todo caracterizando o final de um processo de negócio completo e indivisível. É esse o caso nessa situação em análise?”.

Em caso positivo, cabe confirmar a veracidade da resposta com uma pergunta de confirmação:

“Outros usuários terem acesso aos dados salvos nos passos intermediários é um problema violando as regras que governam o seu sistema? No caso do pedido, antes de se concluir a montagem do pedido existem mecanismos que impedem que esses dados sejam publicados para outros usuários. É esse o seu caso?”.

Se a resposta for positiva, está confirmado que a única razão para as salvagens intermediárias são os requisitos de confiabilidade do sistema. A pergunta de confirmação, muitas vezes apresenta um **“falso negativo”**, na medida em que a resposta pode ser “tanto faz” e como sua elaboração e desenvolvimento envolve um trabalho adicional, muitas vezes nem é dada essa

opção ao usuário que não se apercebe do impacto negativo do atendimento desse requisito não funcional nas suas práticas e procedimentos (funcional).

### **3.3. Orientações para Distinguir Requisitos de Desempenho**

---

Ao identificar arquivos responsáveis por manter dados consolidados, deve-se avaliar se esse requisito é relativo à viabilização de maior desempenho (requisito não funcional) ou relativo às práticas e procedimentos do negócio. Para identificar isso, faça a seguinte pergunta ao especialista no assunto:

“Se esses dados consolidados não fossem mantidos pelo sistema, ainda assim ele funcionaria? Não importa se demorasse uma eternidade para que uma resposta seja fornecida pelo sistema... usando os dados analíticos que subsidiaram a criação desses dados consolidados, os resultados atenderiam às suas necessidades?”.

Em caso positivo, cabe confirmar a veracidade da resposta com uma pergunta de confirmação:

“Sempre que um dado analítico utilizado na geração desse dado consolidado em análise, é atualizado (alterado ou excluído), isso se reflete no dado consolidado? Eles sempre são mantidos em sincronia? Quando os dados analíticos são expurgados, os dados consolidados refletem isso?”.

Se a resposta for positiva, está confirmado que a única razão para manter esses dados consolidados seja o atendimento de requisitos de desempenho e, portanto, não é um requisito funcional.

### **3.4. Orientação Geral**

---

Um requisito funcional é algo particular e específico de uma tarefa ou serviço; contrastando com um requisito geral ao sistema como um todo. Alguns exemplos disso são explícitos no manual de contagem e há orientação explícita para que sejam desconsideradas na análise como, por exemplo:

- Haver em todas as páginas com consultas do sistema informação como número de página;
- Informações de datas em cabeçalhos;
- Informação de posicionamento como registro 37 ao 54 de 211.

Mesmo havendo um caso de uso ou outra especificação de requisitos descrevendo um alegado requisito funcional, verifique se está claro quando um requisito é funcional e quando ele é um requisito não funcional. Considere apenas os primeiros em sua análise.

#### **4. Fator de Ajuste**

---

O FNDE **não** utiliza a avaliação do **valor do fator de ajuste** em suas medições, sendo este fixado no valor 1,00. Ou seja, as características gerais do sistema, apresentadas no manual de práticas de contagem do IFPUG **não** afetam o tamanho dos Pontos de Função **Não-Ajustados** obtidos em suas contagens, fazendo com que seus valores sejam **sempre** iguais aos dos Pontos de Função **Ajustados**.

## 5. Deflator Conforme o Tipo de Ação na Função

A Análise de Pontos de Função (conforme exposta pelo IFPUG) **não** faz nenhum tipo de diferenciação com relação ao **tipo de ação** realizada sobre a funcionalidade (inclusão, alteração ou exclusão) em um projeto de melhoria. A APF não mede o impacto da mudança na função, apenas que a função será alterada e o tamanho que a mesma terá após a alteração.

No entanto, ao se utilizar a técnica para realização de **pagamentos e estimativas de esforço** é necessária uma **abordagem diferenciada** com intuito de permitir uma relação mais próxima entre o tamanho funcional e o esforço estimado ou remuneração efetuada.

O **esforço** despendido durante a construção de uma **nova função**, via de regra, não é o mesmo despendido para realização de uma **manutenção** na funcionalidade, assim como a de **exclusão** da mesma. Desta forma, a solução utilizada é a adoção de **deflatores** conforme o **tipo da ação** executada sobre a função, **diminuindo** a quantidade de PF identificados para as funcionalidades **alteradas e excluídas** para fins de pagamento e estimativas de esforço.

Esta é uma abordagem comum no mercado e vem sendo utilizada por praticamente todos aqueles que utilizam a técnica da APF para os fins citados anteriormente. Como exemplo, temos o Banco do Brasil, CEF, BACEN, dentre outros.

No âmbito do FNDE, os seguintes deflatores serão utilizados, conforme o tipo de ação executada sobre a funcionalidade:

<b>Tipo de Ação:</b>	<b>Deflator:</b>
Funções incluídas (novas)	1,0 (100%)
Funções alteradas	0,6 (60%)
Funções excluídas	0,3 (30%)

## 6. Medição das Manutenções

A APF **não** mede todos os tipos de manutenção possíveis em um software, restringindo-se apenas aquelas que alteram os requisitos de **negócio/funcionais**. Estas são chamadas pelo IFPUG de *Manutenções Adaptativas*. No entanto, a terminologia utilizada pelo FNDE se difere da adotada pelo IFPUG, conforme apresentado na tabela comparativa a seguir:

IFPUG	FNDE	Significado
Manutenções <b>Adaptativas</b>	Manutenções <b>Evolutivas</b>	Alteração de requisitos <b>funcionais</b> .
Manutenções <b>Perfectivas</b>	Manutenções <b>Adaptativas</b>	Alteração de requisitos <b>não funcionais</b> .

No caso destas manutenções mensuráveis, a técnica medirá sempre a funcionalidade **completa**, da forma que será entregue ao usuário, independente da extensão da manutenção. Ou seja, mede-se a **função** que foi alterada e **não o quanto** ela foi alterada. Portanto, conclui-se que a granularidade com que a APF mede a manutenção não é muito fina, mas isto não chega a ser um problema. Medir a extensão da manutenção (como é o caso da abordagem da NESMA) seria um refinamento maior da medição, porém tornaria o processo um pouco mais trabalhoso.

Pela maneira como o IFPUG definiu a medição da manutenção em uma função (medir a função toda, como esta será entregue), faz com que, em vários casos, alterações muito pequenas na função tenham o **mesmo** tamanho de uma manutenção extensa na mesma função. Portanto para que não haja desperdício de recursos, é fundamental que se discipline uma maneira mais "racional" na demanda por manutenções (tratado a seguir).

Quando se está em um contexto em que diversas manutenções serão atendidas ao longo do tempo, o esforço ou custo derivado da medição pela APF para uma manutenção específica pode ser **super** ou **subdimensionado**. Mas ao analisar o conjunto das diversas manutenções em um horizonte de tempo maior (ao menos um ano), estas distorções tendem a se compensar se o parâmetro de preço (R\$/PF) ou taxa de entrega (H/PF) foram bem estabelecidos.

### 6.1. Dicas para otimizar o custo das manutenções

É fundamental que se tenha uma atenção especial na gestão às demandas de manutenção de seus sistemas. Se toda demanda de manutenção que chegar for encaminhada

diretamente para execução pelo fornecedor, a tendência é que o **custo** destas manutenções ao final sejam **superiores** ao que poderiam ser, caso houvesse um controle sobre estas solicitações.

A seguir, algumas dicas podem ajudar a melhorar este cenário:

- a) Consolidar manutenções na **mesma função** em uma **única demanda** é a maneira mais fácil de racionalizar o custo.

Já que o IFPUG não mede a extensão da manutenção na função afetada, fazer uma manutenção para atender a um único requisito ou para atender a vários requisitos de manutenção na **mesma** função terá o mesmo tamanho funcional, se elas forem solicitadas para o fornecedor no mesmo momento. Se solicitadas em momentos distintos, as mesmas funções serão pagas **várias** vezes, para cada um dos Projetos de Melhoria.

No entanto, nem sempre é possível represar uma necessidade do usuário para que esta seja agrupada com outras, já que há demandas com prazos críticos. O importante é tentar avaliar ao máximo quais ajustes realmente são críticos e quais não são, visando evitar o cenário descrito acima.

- b) Reutilizar funções existentes em outros sistemas.

Muitas vezes algumas funções já existem em outros sistemas e mesmo assim ocorre a replicação destas funcionalidades, principalmente com os requisitos de armazenamento (Arquivos Lógicos).

### **Exemplo:**

Suponha que vários sistemas realizam cálculos de impostos a serem pagos, mantendo em sua base de dados local uma tabela com as alíquotas dos impostos e as respectivas funcionalidades de manutenção destes dados. Por conta de um acréscimo no IOF, diversos sistemas tiveram que sofrer manutenção, em alguns deles pagou-se pelas mesmas funções em várias aplicações: o **ALI** que armazena as alíquotas e as transações de **inclusão, alteração e consulta** destes dados.

Se estas funcionalidades estivessem **centralizadas** em um único

sistema, as funções de manutenção das alíquotas de impostos (inclusão, alteração e consulta) seriam contadas no projeto de melhoria do sistema responsável por estes dados e os demais contariam apenas um **AIE** alterado e suas **funções específicas** impactadas, que seriam contadas de qualquer forma

Avaliando apenas as novas funcionalidades e supondo que 4 sistemas fossem afetados, que o Arquivo Lógico fosse de complexidade baixa (7 PF) e que as transações fossem de complexidade baixa (3x 3 PF), para o primeiro cenário haveria um total de 64 PF (1 ALI + 2 EE + 1 CE para cada um dos 4 sistemas), enquanto que no segundo, 31 PF (1 ALI + 2 EE + 1 CE no sistema central e + 1 AIE para cada um dos 3 sistemas restantes). Ou seja, resultaria em uma redução de aproximadamente **50%** no total de PF.

- c) Análise crítica dos requisitos.

Em muitas situações é possível ter uma única função que faça o papel de duas existentes. Isto é muito comum no caso de consultas e relatórios com diferença apenas de alguns atributos apresentados. Ou seja, uma transação mais completa poderia ser elaborada para evitar a criação de várias funções distintas, porém semelhantes.

É mais barato pagar pela criação de uma função nova do que por uma manutenção em duas ou mais funções, principalmente a longo prazo.

## **6.2. Campos Não Editáveis (*Read-Only*)**

---

Para telas de pesquisa, cadastro e alteração que apresentem automaticamente dados preenchidos em campos não editáveis, é importante avaliar a sua necessidade funcional.

### **6.2.1. Cenário I: Campo preenchido automaticamente para restringir valores de entrada do usuário, de acordo com o perfil do usuário autenticado.**

---

The screenshot shows the FNDE web application interface. At the top, there is a header with the FNDE logo and the text 'Fundo Nacional de Desenvolvimento da Educação'. Below this is a navigation bar with 'Menu' and 'SISTEMA DE GESTÃO DO PROGRAMA CAMINHO DA ESCOLA :: VERSÃO. 2.9.0'. The user is identified as 'Usuário CE\_1 | Sair'. The main content area is titled 'Solicitar Adesão ao Registro de Preço' and contains a 'FILTRO' section with the following fields: 'UF' (pre-filled with 'RJ'), 'Tipo de Entidade' (pre-filled with 'PREFEITURA'), 'Entidade' (pre-filled with 'PREF MUN DE RIO DAS OSTRAS'), 'Exercício' (pre-filled with '2009'), and 'Situação da Adesão'. There is also a 'Tipo de Adesão' dropdown menu and a 'Filtrar' button. The browser's address bar shows 'Caminho da Escola - CS001'.

Caminho da Escola - CS001

No exemplo apresentado, os campos *UF*, *Tipo de Entidade* e *Entidade* são carregados e preenchidos automaticamente, caso o usuário autenticado no sistema não possua perfil de Gestor. Estas informações são mantidas em uma entidade externa ao sistema e recuperadas conforme o usuário específico.

Estes são campos do negócio que atravessam a fronteira da aplicação e são utilizados como parâmetros para a funcionalidade de filtro e, portanto, **devem** ser contados como *Tipos de Dado* da funcionalidade em questão.

Vale ressaltar que o fato destes campos serem carregados automaticamente **não** é suficiente para identificação de um segundo *Processo Elementar* para os usuários com perfil de Gestor, uma vez que trata-se de uma facilidade na restrição do escopo da pesquisa.

### 6.2.2. Cenário II: Campos preenchidos automaticamente em uma consulta implícita para alteração de dados

Em situações como estas, é importante avaliar quais informações apresentadas e não editáveis são realmente importantes para a funcionalidade de alteração. Ou seja, informações que não sejam essenciais para a alteração em si, porém são carregadas anteriormente por uma consulta implícita **não** devem ser contadas como *Tipos de Dado* da transação de alteração.

Uma dica é separar a funcionalidade de consulta da funcionalidade de alteração. Imaginar uma tela específica de "Detalhe", onde todas as informações são apresentadas, e uma segunda tela de alteração onde há apenas os campos essenciais e os editáveis. Desta forma, as informações não essenciais à alteração ficariam apenas na tela de "Detalhe" enquanto que os campos não editáveis, porém necessários para a transação, não poderiam ser movidas para esta tela de detalhe, permanecendo na tela de alteração.

### 6.2.3. Cenário III: Campos de cabeçalho apresentados ao longo de todo o sistema

Antes de classificar tal item, é necessário verificar se a apresentação destas informações no cabeçalho atende realmente a requisitos de **negócio** da aplicação ou se são apenas requisitos **não funcionais** da mesma.

Partindo do princípio que a apresentação dos dados no cabeçalho seja uma necessidade do negócio da aplicação, deve-se observar se o cabeçalho é apresentado independente da funcionalidade acessada, assim como a funcionalidade específica acessada não depende necessariamente do cabeçalho para ser significativa para a aplicação.

Ou seja, havendo uma **independência** entre tais itens, um *Processo Elementar* específico deve ser identificado para a capacidade de apresentação de dados do cabeçalho, sendo que seus *Tipos de Dado* e *Arquivos Referenciados* não devem ser identificados nas demais funcionalidades apenas pelo fato do cabeçalho estar sendo apresentado na mesma tela.

### **6.3. Subdivisão de uma Funcionalidade em $n$ Funções**

---

Para correta quebra de uma funcionalidade em diversos *Processos Elementares*, é importante avaliá-las a partir de uma perspectiva do **negócio**, verificando quais funcionalidades são completas e reconhecidas pelos usuários do negócio.

#### **6.3.1. Cenário I: Mais de um *Processo Elementar* identificado para a funcionalidade**

---

MEC
Usuário: Nome
1.4.13
Brasil

## SGD – Sistema de Gestão de Demandas

Menu
Sair

SGD > Demandas > Listar Demandas
Ajud

Demandas
Ordens de Serviços
Atendimentos

Pesquisar

Lista das demandas

ID	Título	Tipo	Prioridade	Situação	Cronograma
<input type="checkbox"/> 701	Versão SGD 2.0	Evolutiva	Alta	Aberta	A definir
<input type="checkbox"/> 702	-	Corretiva	Média	Em andamento	Em andamento
<input type="checkbox"/> 703	-	Evolutiva	Alta	Aguardando Homologação	Em andamento

Voltar
Homologar
Cancelar
Suspender

SGD 2.0 – Listar Demandas

No exemplo apresentado, é possível que o usuário homologue, cancele ou suspenda um conjunto de demandas específicas, selecionadas a partir da listagem ilustrada.

É importante avaliar se há uma **única Função Transacional** para todas as três opções (Ex: Avaliar Demanda) ou se cada um dos itens deve ser classificado como um *Processo Elementar* distinto.

Mesmo que o sistema não dispere nenhum controle específico para cada uma das ações, analisando a partir de uma ótica do negócio, é claro para o usuário que estas são atividades distintas que refletem em situações diferentes para o negócio da aplicação.

Desta forma, três funções transacionais devem ser identificadas, já que é claro para o usuário a distinção do ato de homologar uma demanda, para o de cancelar e suspender, e vice-versa.

### 6.3.2. Cenário II: Um único *Processo Elementar* identificado para a funcionalidade

LSE – Cadastrar escola e anexo – Folha 1

LSE – Cadastrar escola e anexo – Folha 2

No exemplo apresentado, é disponibilizado ao usuário uma funcionalidade responsável por cadastrar escolas. No entanto, o cadastro é dividido em duas telas, chamadas de “folhas”. Em situações como esta, é necessário avaliar se cada uma das telas constitui um *Processo Elementar*, atendendo principalmente ao requisito de ser **auto-contido**.

Neste caso, a primeira coisa a se fazer é a de tentar identificar que razão levou à divisão da funcionalidade em diversas telas. É comum que formulários de cadastro sejam quebrados em etapas com objetivo apenas de tornar a atividade de cadastro mais intuitiva e organizada, ou seja, apenas para atender a requisitos **não funcionais**. Assim sendo, apenas **um** único PE pode ser identificado.

Caso contrário, se a funcionalidade tenha sido subdividida em diversas telas para atender a uma necessidade do **negócio** (Ex: Um departamento possui a competência para o preenchimento de uma das telas enquanto que outro departamento para as demais), será caracterizado **mais de um PE**.

Duas dicas gerais para auxiliar o analista de métricas em situações semelhantes são:

- Verifique se, caso a funcionalidade não fosse fragmentada e houvesse uma única tela, haveria algum impacto no **negócio** da aplicação ou se a funcionalidade se tornaria apenas **menos usual**.
- Verifique se há usuários de **áreas de negócio distintas** responsáveis por preencher telas específicas da funcionalidade, não tendo competência (mesmo que munidos de todas as informações necessárias) para o preenchimento completo do formulário.

Voltando ao exemplo do LSE e aplicando estas duas dicas é possível verificar que existe apenas **um único** PE. O usuário não reconhece como uma atividade do negócio apenas preencher uma folha específica do formulário como um processo auto-contido. O usuário visualiza o cadastro de escola como um todo, porém dividido em duas abas para atender a requisitos **não negociais** (Ex: Layout obrigatório).

#### **6.4. Impressão de Termos e Formulários**

---

Para correta contagem dos *Tipos de Dado* em funções responsáveis pela emissão de termos e formulários, é importante verificar se estes possuem campos recuperados de **Arquivos Lógicos Internos** ou **Arquivos de Interface Externa** (*Arquivos Referenciados* da função) ou até mesmo gerados dinamicamente.

##### **6.4.1. Cenário I: Formulário composto por dados resgatados de vários Arquivos Lógicos**

---

Ofício nº \_\_\_\_\_

RIO DAS OSTRAS - RJ, 4 de abril de 2008.

A Sua Senhoria o Senhor  
DANIEL SILVA BALABAN  
Presidente do Fundo Nacional de Desenvolvimento da Educação - FNDE  
SBS Quadra 02 Bloco F Edifício Áurea  
Brasília-DF

Assunto: Adesão a Ata de Registro de Preços referente ao Pregão Eletrônico nº \_\_\_\_/200\_

Senhor Presidente,

Com fulcro no art. 8º, § 1º, do Decreto nº 3.931, de 19 de setembro de 2001, consulto  
Vossa Senhoria sobre a possibilidade de adesão à Ata de Registro de Preços referente ao Pregão Eletrônico  
nº \_\_\_\_/200\_, realizado pelo Fundo Nacional de Desenvolvimento da Educação - FNDE.

A referida adesão visa à aquisição dos itens abaixo demonstrados:

Composições	Qtde Total Composição	Qtde Adaptado	Valor Total Solicitado
1) Um onibus de 44 passageiros	4	0	R\$ 690.800,88
2) Um onibus de 31 passageiros	1	0	R\$ 126.750,44
3) Um onibus de 23 passageiros	1	0	R\$ 114.000,00
4) Uma embarcacao de 35 passageiros	1	-	R\$ 300.000,00
5) Uma embarcacao de 20 passageiros	0	-	R\$ 0,00
6) Dois onibus de 23 passageiros	0	0	R\$ 0,00
7) Duas embarcacoes de 20 passageiros	0	-	R\$ 0,00
8) Um onibus de 23 passageiros e uma embarcacao	0	0	R\$ 0,00

**Valor Solicitado: R\$ 1.231.551,32**

Caminho da Escola - CS001

No exemplo apresentado, existem tanto campos resgatados de *Arquivos Lógicos* do sistema, como um dado derivado (*Valor Solicitado*).

Portanto, deve-se identificar em quais Arquivos Lógicos os campos resgatados são armazenados para que estes sejam identificados como **Arquivos Referenciados** da função.

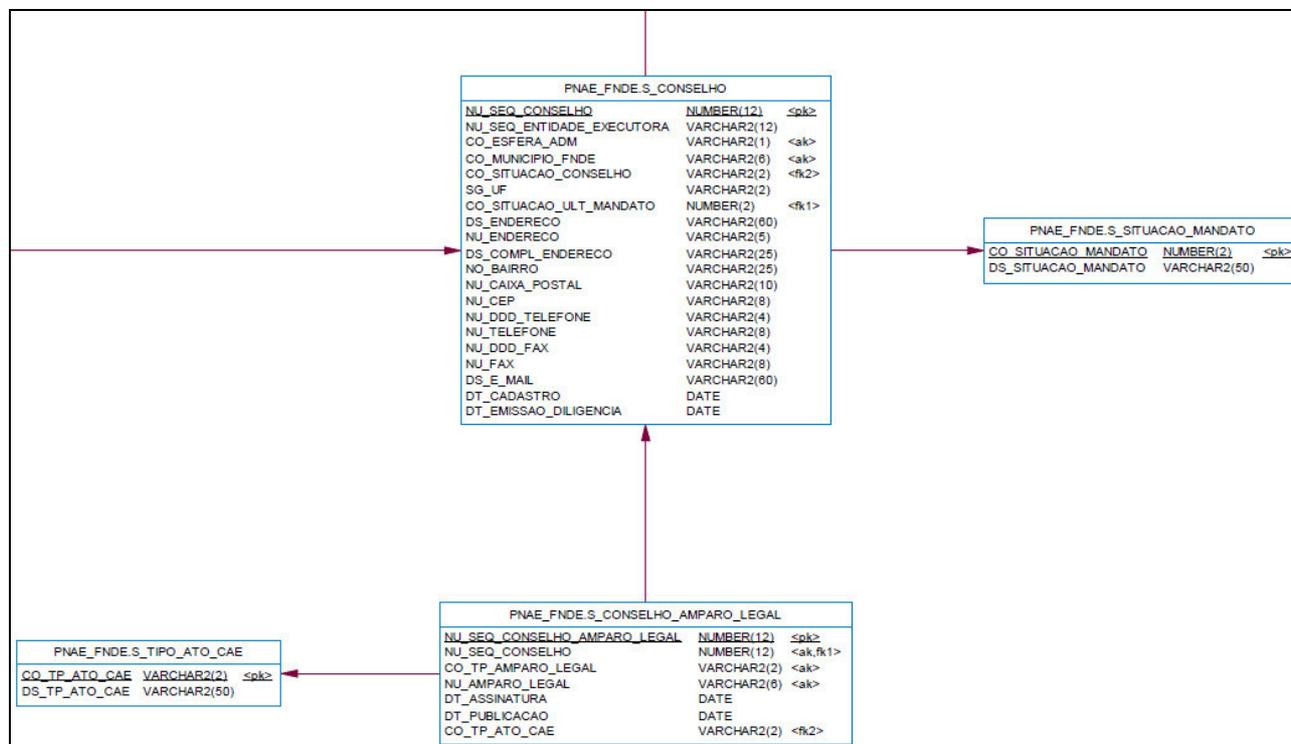
## 6.5. Envio de Email

Muitas funções possuem embutidos em seu processamento, o envio de emails de notificação ao final.

### 6.5.1. Cenário I: Envio de e-mail como consequência de um processo



Para correta contagem dos *Tipos de Registro* e *Arquivos Lógicos*, é importante verificar as entidades do negócio e suas respectivas **dependências funcionais**.



CAE Virtual - Fragmento MER

No exemplo são apresentadas quatro tabelas distintas, porém apenas um único *Arquivo Lógico* pode ser identificado, este com apenas dois *Tipos de Registro*.

Note que as tabelas SITUACAO\_MANDATO e TIPO\_ATO\_CAE são compostas apenas por código e descrição, sendo um exemplo clássico de *Dados de Código* e, portanto, **não** podem ser reconhecidas como *Arquivos Lógicos* ou *Tipos de Registro* de um *Arquivo Lógico*.

Já a tabela CONSELHO\_AMPARO\_LEGAL, a partir do ponto de vista do negócio da aplicação, representa uma entidade do negócio **dependente** da entidade descrita pela tabela CONSELHO. Esta conclusão pode ser obtida ao analisarmos as regras e procedimentos do negócio do sistema, assim como a forma pela qual as informações mantidas em ambas as tabelas são cadastradas e atualizadas, a partir da **visão do usuário**. Ou seja, ambas constituem um **único Arquivo Lógico** e este por sua vez é composto por dois *Tipos de Registro*.

Note que não há requisitos transacionais responsáveis por manter especificamente as informações armazenadas na tabela CONSELHO, isoladamente às informações mantidas na tabela CONSELHO\_AMPARO\_LEGAL (e vice-versa). Este é um forte indício de que a separação feita no modelo de dados ocorreu apenas para atender

a requisitos **não funcionais** (Ex: Normalização). Indício que se confirma ao analisarmos a relação de dependência entre estas informações perante uma ótica do **negócio**, sem levar em consideração a solução técnica e observando apenas a forma pela qual o sistema trata estes dados com o usuário.

A seguir é apresentada a tela de cadastro responsável por cadastrar dados em ambas as tabelas:



The screenshot shows the FNDE (Fundo Nacional de Desenvolvimento da Educação) interface for the 'CONSELHO DE ALIMENTAÇÃO ESCOLAR'. The form is divided into several sections:

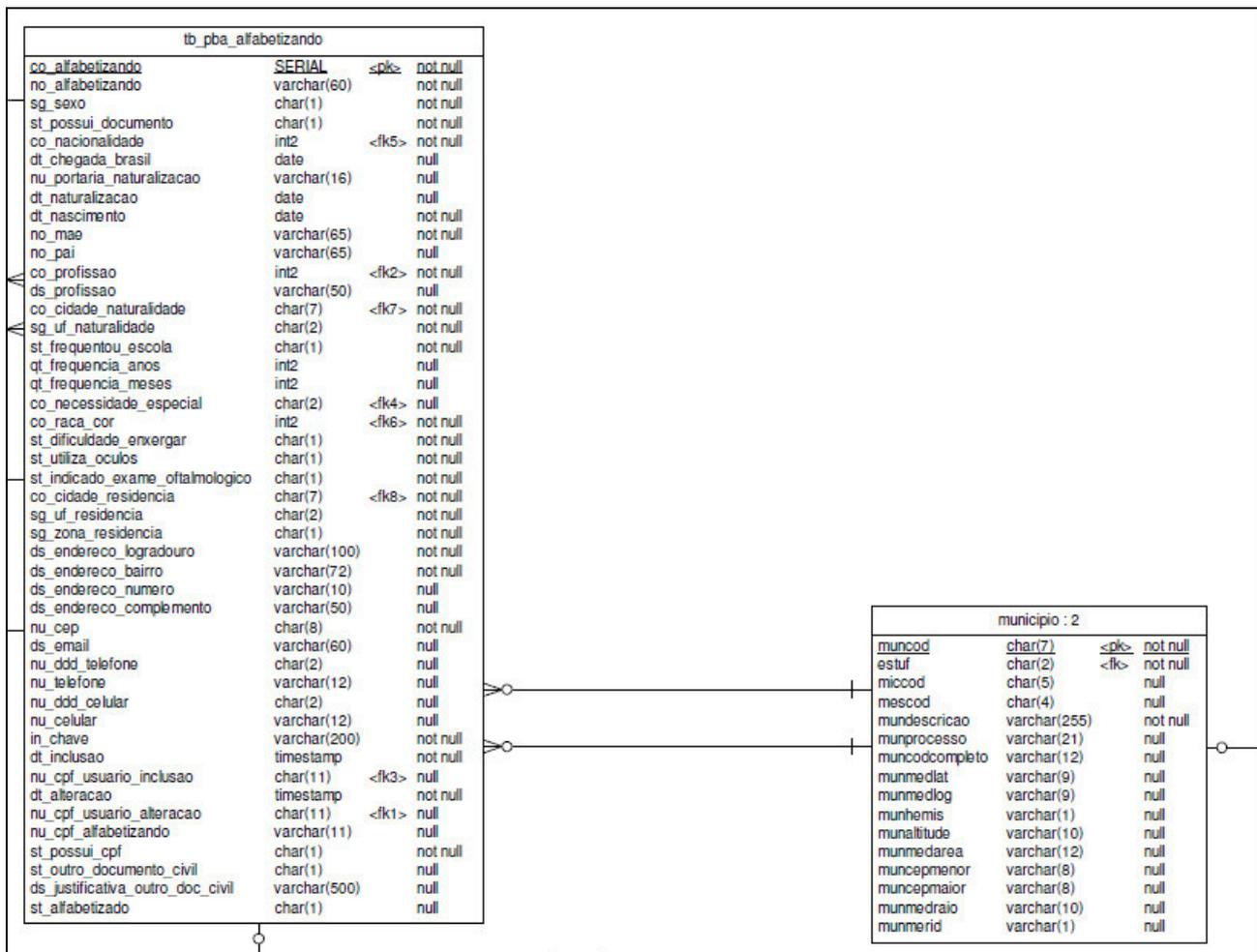
- Dados Cadastrais do Conselho:** Includes fields for 'Endereço (rua, avenida ou praça)' (RUA GERALDO BARBOSA), 'Complemento do Endereço (andar, sala, etc)', 'Número' (485), 'Bairro / Distrito' (CENTRO), 'UF' (ACRELANDIA), 'Município' (ACRELANDIA), 'CEP' (69945000), 'Caixa Postal', 'DDD' (68), 'Telefone' (32351430), 'DDD' (68), 'Fax' (32351430), and 'E-mail' (handoboliva@hotmail.com).
- Ato de Criação do CAE:** Includes 'Tipo de Amparo Legal' (DECRETO MUNICIPAL), 'Número' (22), 'Data de Assinatura' (03/04/2001), 'Data de Publicação' (03/04/2001), and 'Data do Cadastro' (28/05/2001).
- Ato de Regulamentação do CAE:** Includes 'Tipo de Amparo Legal', 'Número', 'Data de Assinatura', and 'Data de Publicação'.
- Última Alteração do Ato de Criação do CAE:** Includes 'Tipo de Amparo Legal', 'Número', 'Data de Assinatura', and 'Data de Publicação'.

At the bottom of the form, there is a red note: '(\*) Preenchimento Obrigatório' and buttons for 'Voltar' and 'Atualizar'.

CAE Virtual – Manter Dados CAE

Conforme ilustrado, existe apenas uma única tela encarregada pelo cadastramento de tais informações. Os dados referentes ao conselho em si (parte superior) e ao amparo legal (parte inferior da tela) são dependentes, sendo reconhecidas como um único requisito de armazenamento de dados para o negócio da aplicação.

É necessária uma atenção especial com relação à distinção dos conceitos de *Dependência Funcional* e *Obrigatoriedade de Relacionamento*. Observe que o último restringe associações dentro de um modelo de dados, ou seja, uma entidade deve obrigatoriamente apontar para um registro em outra entidade, porém isto **não é suficiente** para declará-las como dependentes funcionais entre si. Uma entidade pode ter significado para o negócio independente de qualquer outra, mesmo que esta possua diversas obrigadoriedades de relacionamento (restrições de FK), conforme apresentado a seguir:



SBA – Fragmento MER

Neste caso, a entidade do negócio *Alfabetizando* **não** depende funcionalmente de um município para existir. Este é apenas um de seus atributos que aponta para outra entidade do negócio desta aplicação e, portanto, *Arquivos Lógicos* distintos devem ser identificados.

## 6.7. Conversão de Dados

Para correta medição das funções de conversão, é importante verificar se a atividade relacionada realmente deve ser classificada como uma função de conversão de dados ou se trata apenas de uma **atividade de suporte**, que não pode ser classificada como uma **função** (vide mais detalhes a seguir).

As funções de conversão de dados devem possuir dados **atravessando** a fronteira da aplicação e serem **descartáveis** (executadas uma única vez em produção). Este tipo de funcionalidade possui a intenção primária de manter um ou mais ALIs da aplicação e, portanto, são classificadas como **Entradas Externas**.

Também é importante atentar-se à análise correta dos relatórios comumente emitidos por estas funcionalidades. Estes são **conseqüência** da atividade de migração, não sendo completos em si (auto-contidos) e, portanto, **não** devem ser medidos como uma nova função, independente da migração em si.

A dica geral é fazer a **analogia** entre a funcionalidade de conversão com a atividade de migração de registros de forma manual (usuário entrando com dados por meio de uma tela de cadastro). Se não houvesse a funcionalidade, o usuário informaria os campos na tela (**dados entrando pela fronteira**) e faria toda esta atividade uma única vez (**descartável**), já que, após concluída a migração, os registros passariam a ser cadastrados conforme o negócio da aplicação. Ou seja, a função de conversão de dados é responsável apenas por **automatizar** a execução de toda a atividade de entrada de dados, caracterizada pela busca dos dados que entrarão pela fronteira da aplicação em alguma fonte externa (Ex: planilhas, tabelas, arquivos txt) e pelo fato de ser descartável, já que não terá mais utilidade após a sua execução.

#### **6.7.1. Cenário I: Migração de dados dissociada de projeto**

---

O Sistema de Consulta de Distribuição do Livro é uma pequena aplicação de apoio à operação da distribuição de materiais didáticos entre as escolas. Porém este será descontinuado e substituído por um **novo módulo** no próprio SIMAD – Sistema de Material Didático.

Devido a restrições do negócio, a **migração dos dados** já cadastrados no Sistema de Consulta de Distribuição do Livro **não** será realizada em conjunto com a criação do próprio módulo, sendo necessário aguardar certo período do ano para que os dados sejam migrados de forma consistente e sem impacto na execução das atividades do negócio relacionadas.

Desta forma, em um primeiro momento foi solicitada a criação do novo módulo e posteriormente foi gerada uma nova demanda referente à migração de dados. Neste caso, a migração de dados será identificada normalmente como uma **função de conversão** e deverá ser medida em PF.

O fato de a migração ter sido solicitada isoladamente em nada afeta a análise de Pontos de Função. Esta seria medida da mesma forma caso fizesse parte da demanda que solicitou a criação do novo módulo. O **cuidado** a ser tomado é com relação à **recontagem** da mesma migração como uma função de conversão, caso esta já tenha sido medida e paga anteriormente.

#### **6.7.2. Cenário II: Migrações de dados frequentes/periódicas**

---

**Anualmente** uma aplicação do FNDE deve realizar a **migração/importação** de dados específicos do MEC. Estas cargas são executadas uma única vez ao ano, porém sempre ocorrem no mês de Janeiro, sendo que a cada ano esta carga de dados passa por **alterações** em seus campos, já que estão sujeitos a alterações feitas pelo MEC.

Nesta situação, **não** há função de conversão de dados. As migrações são **executadas periodicamente** (mesmo que em períodos longos de um ano), o que confronta o caráter descartável das funções de conversão. Há na realidade uma **função transacional** responsável pela carga dos dados na aplicação do FNDE, que fatalmente passará por manutenções **funcionais** ao início de cada ano, salvo situação em que esta atividade seja executada diretamente pela equipe de desenvolvimento, sem que haja uma **funcionalidade** responsável por tal tarefa (um exemplo disso é a carga do Senso Escolar, que é executada anualmente pela equipe de banco de dados, não havendo uma funcionalidade para automatizar este processo).

Note que, caso fossem identificadas várias **funções de conversão** ao invés de um **Processo Elementar** sendo modificado - além da distorção conceitual e possível impacto na medição da baseline - nenhum **deflator** seria aplicado a estas funções.

## 6.8. Middleware

---

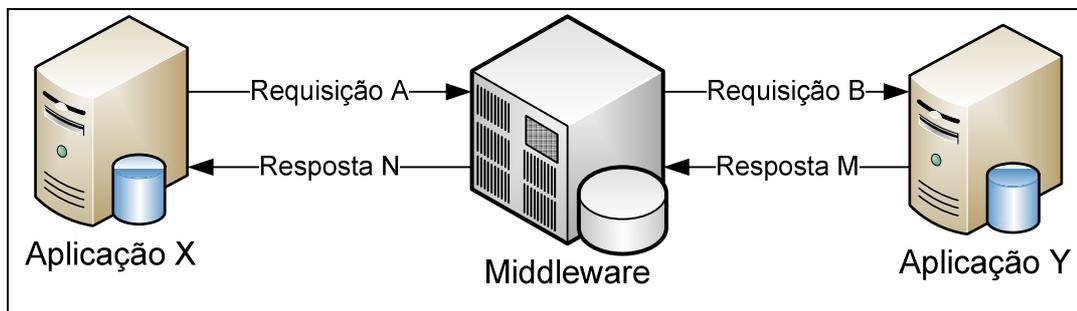
Para correta medição de aplicações de *middleware*, é essencial avaliar e definir corretamente a sua **fronteira** e os **usuários** que interagem com o sistema. Como o **negócio da aplicação** é geralmente servir como um **intermediário** para comunicação entre dois ou mais sistemas, caso a fronteira do *middleware* não esteja clara, esta poderá englobar parte das fronteiras das aplicações responsáveis por interagir com o sistema, comprometendo a medição realizada.

Da mesma forma, caso os usuários da aplicação não sejam identificados corretamente, a quantidade de funções identificadas não representará a realidade. Se houver usuários que não foram reconhecidos durante a medição, a quantidade de funções identificadas para aplicação será **inferior** à quantidade real, enquanto que se forem identificados usuários inexistentes, diversas funções serão identificadas erroneamente.

### 6.8.1. Cenário I: Sistema intermediário

---

Dentre as várias implementações possíveis para este tipo de solução, a mais comum é aquela em que o sistema é utilizado para conectar duas outras aplicações separadas. Assim, os usuários de *middlewares* são, geralmente, outras aplicações.



Exemplo Middleware

Neste exemplo, duas aplicações, X e Y, precisam trocar informações. Porém, elas não utilizam a mesma linguagem, precisando de uma aplicação intermediária (*middleware*) para que elas possam se comunicar. Deste modo, conforme figura acima, a Aplicação X envia uma Requisição A para o *middleware* e este a traduz, registra no log e a envia à Aplicação Y, de modo que Y consiga interpretar e processar. A aplicação Y processa esta mensagem e retorna uma Resposta M ao *Middleware* para traduzi-la e enviá-la à Aplicação X, para que este processe o retorno e conclua a operação.

### Fronteira da Aplicação:

Com base no cenário apresentado, é possível verificar que as aplicações em interface com o *middleware* são os **usuários**. Eles, logicamente, reconhecem as traduções que são enviadas e recebidas. Desde que o propósito seja medir o tamanho da aplicação de **middleware**, a fronteira é limitada para as funcionalidades fornecidas pelo sistema intermediário.

### Arquivos Lógicos Internos:

De acordo com os requisitos, todas as transações recebidas e enviadas para o roteador são **registradas** no arquivo de log do roteador. Com isso, pode-se concluir que o log do roteador é um **ALI** para a aplicação do *middleware*.

### Arquivos de Interface Externa:

Como nenhum *Arquivo Lógico* externo à fronteira do sistema de *middleware* é referenciado, **nenhum** AIE pode ser identificado.

### Entradas Externas:

**Nenhuma** EE pode ser identificada, pois todas as transações têm intenção de enviar dados para fora da fronteira do *middleware*.

Saídas Externas:

Os requisitos do usuário identificam as seguintes funcionalidades:

- Conversão e envio de uma mensagem da Aplicação X para a Aplicação Y.
- Conversão e envio de uma mensagem da Aplicação Y para a Aplicação X.

Consultas Externas:

**Nenhuma** CE pode ser identificada, pois todas as transações mantêm o *log* do roteador.

### **6.8.2. Cenário II: Webservice dentro da fronteira da aplicação**

---

Em uma aplicação há a necessidade de **disponibilizar informações** específicas do negócio do sistema para outras **aplicações**. A fim de atender a esta necessidade, foi implementado um **Webservice** encarregado de publicar as informações pertinentes aos seus respectivos sistemas. Esta opção foi tomada pelo gerente responsável pelo projeto com intuito de facilitar futuras manutenções e o acesso externo dos demais sistemas.

Note que, para o cenário descrito acima, o Webservice em si **não** é caracterizado como uma aplicação **distinta**, com uma **fronteira própria**. Este é apenas a solução técnica adotada para disponibilizar os dados necessários às diversas aplicações. Inclusive, outras soluções poderiam ter sido utilizadas, sem que houvesse qualquer tipo de impacto nos requisitos **funcionais** da aplicação (Ex: tabelas compartilhadas; arquivos txt; acesso direto a base de dados; etc).

A correta definição da fronteira é **essencial** e merece especial cuidado. Observe que, ao **contrário** do cenário apresentado no tópico anterior (6.8.1) onde há uma fronteira própria para a aplicação de middleware (cujo **negócio/objetivo** é justamente servir como uma aplicação **intermediária** entre um ou mais sistemas), neste caso o Webservice foi implementado apenas com intuito de atender a uma **necessidade específica** da aplicação como um todo. O Webservice é apenas um dos componentes técnicos utilizados pela aplicação, cujo respectivo negócio **não** é servir como intermediária entre outros sistemas.

No que diz respeito à contagem, para cada um dos **sistemas e conjunto de informações** disponibilizados, haverá (desde que atenda as respectivas regras de identificação, expostas no Manual de Práticas de Contagem) uma **função transacional** do tipo **CE** ou **SE**.

No entanto, caso a **intenção primária** da função não seja a de disponibilizar informações a outras aplicações, e sim a de **manter** um ou mais **Arquivos Lógicos** ou **alterar o comportamento** do sistema, a função deverá ser classificada como uma **EE**. Nestas situações, a transação funciona como uma transação de cadastro, porém disponibilizada via Webservice.

Para correta identificação das transações, a dica geral é fazer uma **analogia** das informações sendo disponibilizadas com **telas de relatórios** comuns, acessadas por **usuários finais** (pessoas e não sistemas). Ou seja, se o usuário que necessita da informação não fosse outra aplicação, a mesma funcionalidade poderia ser desenvolvida como um relatório em que todas as informações necessárias seriam apresentadas.

Lembre-se: as regras de identificação e unicidade são as **mesmas**, porém adotando uma solução (técnica) específica. Utilize a dica descrita no parágrafo anterior durante a realização da contagem, evitando a identificação **errônea** de vários Processos Elementares para uma única consulta e vice versa.

## **6.9. Processos Sem Interface Gráfica (batch)**

---

Para correta medição de rotinas puramente batch, é importante verificar quais **usuários** estão interagindo com o sistema durante a execução da rotina e se há dados passando pela **fronteira da aplicação**, seja entrando ou saindo.

É comum a supressão deste tipo de rotinas em contagens, alegando que este tipo de processamento simplesmente não é passível de contagem. No entanto, esta é uma inverdade que pode resultar em um impacto drástico na medição final de uma demanda. O fato de uma rotina não possuir interface gráfica de entrada ou saída com a pessoa do negócio que utiliza o sistema **não** é suficiente para descartar a funcionalidade durante a medição.

O conceito de *Usuário* para a APF não se restringe apenas a pessoas físicas, englobando também outros **sistemas** que se comuniquem com a aplicação sendo analisada. Ou seja, rotinas batch sem interação humana, mas que são responsáveis pelo tráfego de informações entre sistemas distintos (que é um cenário comum), **podem** ser medidas, desde que atendam às regras de identificação de um *Processo Elementar*.

Ao se deparar com este cenário, o primeiro passo que o analista deve realizar é identificar os possíveis **usuários** que estejam se comunicando com o sistema sendo medido, por meio desta rotina batch. Se não houver **nenhum** usuário (Ex: rotina que ordena os registros cadastrados no dia por ordem alfabética, sem emitir nenhum tipo de saída), **não** há *Processo Elementar*. A APF é baseada na visão do **usuário**, o que

impede que haja uma funcionalidade totalmente interna, sem nenhum tipo de interação com ao menos um usuário (seja uma pessoa física ou outro sistema).

Havendo um usuário interessado na execução da rotina, é necessário verificar se esta atende às **regras de identificação** de um PE, assim como em qualquer outra funcionalidade transacional. A dificuldade pode estar em entender a lógica da funcionalidade que possui apenas outro sistema como usuário para que a atividade seja executada de forma **automatizada**, porém a análise é exatamente a mesma.

A dica geral é imaginar que o usuário **sistema** não existisse e o processo fosse manual. Ou seja, ao invés de outra máquina interagindo via rotina batch, haveria uma **pessoa** entrando ou recebendo dados por meio de telas, tornando a análise da funcionalidade mais simples.

### 6.9.1. Cenário I: Rotina Batch Não-Mensurável

---

A implementação de rotinas de carga/limpeza de banco de dados programadas para serem executadas periodicamente e, normalmente, fora do horário comercial, é uma prática comum em qualquer instituição.

Não havendo nenhum tipo de relatório ou e-mail sendo disparado como consequência a execução da rotina, **não** há funcionalidade passível de medição, uma vez que não existem dados atravessando a **fronteira da aplicação**.

Note que mesmo sendo mantido um registro em uma tabela de *Log*, por exemplo, a análise continua a mesma. A atividade de registrar uma informação em um ALI **não caracteriza** a entrada ou saída de dados pela fronteira, até porque, como o próprio nome diz, este é um arquivo **interno** ao sistema.

O mesmo ocorre com a existência de dados sendo resgatados de AIEs. O fato de uma transação possuir um AIE como arquivo referenciado **não caracteriza** a entrada ou saída de dados pela fronteira. Lembre-se: os dados devem entrar ou sair para **usuários** do sistema.

### 6.9.2. Cenário II: Rotina Batch Mensurável

---

Agora supondo uma situação, ao final do dia, o sistema execute uma rotina batch responsável por verificar todos os processos administrativos registrados no dia e disparar um e-mail às partes envolvidas.

Note que neste caso **há** dados atravessando a fronteira. O e-mail se assemelha a um **relatório**, onde o usuário consultaria todos os novos processos administrativos que o envolva, por meio de telas. A diferença é que a atividade de verificação é automatizada e o relatório é disparado via e-mail aos interessados.

### 6.9.3. Cenário III: Rotina Batch como parte de outros Processos

---

Outra situação extremamente comum é a que uma funcionalidade é disponibilizada via interface gráfica, porém, do ponto de vista do negócio, esta funcionalidade só é executada parcialmente, sendo consolidada ao final do dia com a execução de uma rotina batch.

Supondo uma situação onde o usuário realize uma transferência bancária, porém esta só seja efetivada ao final do dia por uma rotina batch. Partindo do princípio que a atividade apenas de solicitação **não seja significativa** para o negócio, que só entende esta atividade como completa quando a transferência é efetivamente realizada, o PE é composto de **duas partes**, uma onde o usuário interage diretamente com o sistema, e outra puramente batch.

Como a funcionalidade como um todo é significativa para o usuário, caso a transferência fosse realizada em **tempo real**, em nada afetaria o negócio da aplicação. A quebra da atividade ocorre para atender a requisitos **não-funcionais** (Ex: capacidade de processamento limitada). Ou seja, ao identificar esta funcionalidade, contar apenas os TDs e ARs da parte gráfica está **incorreto**. Durante a execução da rotina, pode haver ARs que não são acionados na primeira etapa da funcionalidade e que **devem** participar da contagem sendo realizada.

Além disso, atente-se à contagem dos TDs. Caso existam campos adicionais sendo trabalhados pela rotina batch, mas que não atravessem a fronteira em momento algum, estes **não** devem ser contados, assim como preconiza as regras de identificação de TDs expostas no CPM.

Note também que neste tipo de solução é comum a utilização de **tabelas/arquivos temporários**, criados apenas para armazenar as informações enquanto a rotina não é executada. Geralmente este tipo de implementação é puramente **técnica** e **não** deve interferir na medição funcional.

### 6.10. Data Warehouse e BI

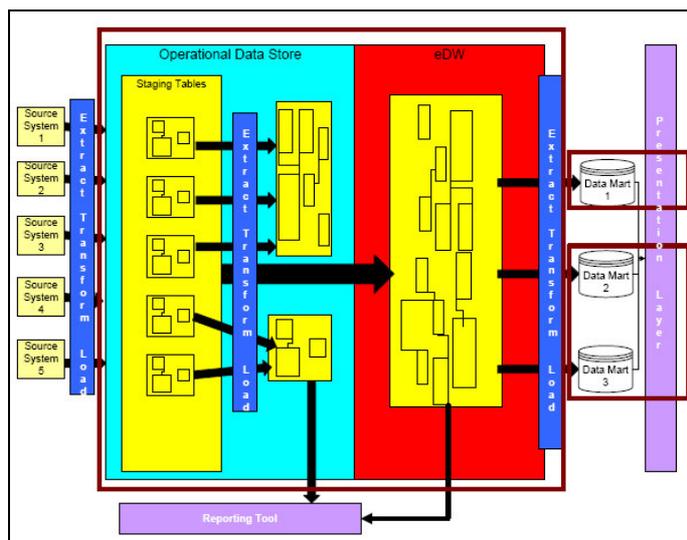
---

Para a correta medição de aplicações de Data Warehouse, é importante ater-se aos conceitos da técnica de Análise de Pontos de Função, evitando vícios de contagem devido à implementação técnica atípica deste tipo de sistema.

A APF mede software, independente de qual seja a tecnologia e da maneira pela qual este tenha sido implementado. Ou seja, por mais peculiar que este tipo de aplicação possa ser, ela **pode** ser medida em PF.

Projetos de Data Warehouse contêm muitos documentos que o Analista de Métricas pode usar para auxiliar na contagem do ponto de função. Alguns dos documentos mais úteis são os diagramas de modelo de dados, que ajudam a determinar funções de dado e de transação.

## 6.10.1. Cenário I: Exemplo de Data Warehouse



Function Points & Counting Enterprise Data Warehouses – Exemplo DW

### Fronteira da Aplicação:

Assim como nas demais aplicações, a definição da fronteira é uma atividade de extrema importância para a medição de DWs. É vital enquadrar todos os componentes próprios da aplicação **dentro** de sua fronteira, deixando claros aqueles elementos que extrapolam a fronteira do DW (Ex: Sistemas de origem de dados e Data Marts).

Caso haja alguma ferramenta/componente web responsável por tornar as informações mais acessíveis aos usuários, este deverá ser analisado com intuito de posicioná-lo corretamente na definição da fronteira. Algumas dicas para ajudar nesta decisão são:

- Se um *Website* central é usado para acessar vários DWs dentro da empresa e este é mantido por uma equipe, essa é uma boa **indicação** de que a ferramenta é uma aplicação **separada**, cuja intenção primária é fornecer acesso aos DWs.
- Se a ferramenta foi construída para fornecer capacidades de relatório para um DW **específico**, e é mantida pela mesma equipe responsável pelo DW, provavelmente, seria contado como parte do **mesmo** aplicativo.

**ATENÇÃO:** A fronteira do aplicativo não se baseia necessariamente em **como** a organização do software é gerenciada. Porém, conhecer quem desenvolveu o componente que fornece acesso a um DW em particular pode ser **útil** quando se define a fronteira da aplicação.

#### Analizando seus Componentes:

Alguns elementos presentes em aplicações de DW são tipicamente **técnicos**, criados para garantir o funcionamento da solução, enquanto que outros possuem significado para o **negócio** da aplicação e, portanto, devem ser levados em consideração durante o processo de medição.

#### Elementos puramente técnicos:

- *Staging Areas*
- Depósito de Dados Operacionais (ODS)
- *Data Warehouse*

#### Elementos com significado para o negócio:

- Tabelas Fato: Caracterizam, em conjunto com as tabelas dimensionais, **ALIs** da aplicação.
- Tabelas Agregadas: Podem ou não ser contadas, dependendo da razão de sua existência. Se tiver sido criada para atender a requisitos do **negócio**, poderá participar da medição como um **ALI**.
- Tabelas de Existência: Podem ou não ser contadas. Se atender a necessidade de documentar a **existência de um evento**, participará da contagem (geralmente como um Arquivo Lógico de complexidade **baixa**).
- Tabelas Dimensionais: Caracterizam, em conjunto com a tabela fato, um **ALI** da aplicação. Ou seja, cada tabela dimensional é um TR do ALI principal, que é a tabela Fato.
- Tabelas de Visão/Visualização: Podem ser identificadas na medição como **CE** ou **SE**, caso sejam criadas com intuito de

fornecer informações para outro aplicativo, fora da fronteira do DW em questão.

- Metadados: Podem participar da medição como **ALIs** caso estejam relacionados à questões negociais da aplicação (Ex: dicionário de dados, proprietários de dados e assuntos sobre a área de informação)
- ETL - Extrair, Transformar e Carregar: Participam da medição como única **EE** para cada ETL.
- Relatórios Ad-Hoc: **Não** participam da contagem de PF, salvo no caso em que haja uma funcionalidade oferecida ao usuário para criar seus próprios relatórios ad-hoc.
- Relatórios Programados: Caso atendam às regras de identificação de um PE, estes relatórios participarão da medição como **CE** ou **SE**.
- Funções Administrativas: Participarão da medição caso a sua natureza esteja relacionada ao **negócio** da aplicação.
- Funções de Metadados: Podem ou não ser contadas. Caso tenham sido desenvolvidas para apoiar um usuário do aplicativo de forma **negocial**, serão identificadas como **PEs**.

#### Tabelas Fato x Tabelas Dimensionais:

Fisicamente, tabelas fato contêm somente os TDs requeridos para representar alguma **medida do negócio** e são rodeadas por tabelas dimensionais que permitem **descrever** um evento em particular.

Para cada conjunto de tabelas fato e suas respectivas dimensões deve-se identificar um **ALI**. Este por sua vez possuirá um **TR** adicional para cada dimensão associada à tabela fato. Note que apenas o **conjunto** (fato + dimensões) é significativo ao usuário.

Uma contagem de TDs tipicamente incluirá campos da entidade da tabela fato **e** da entidade dimensional, no entanto, certifique-se de contar somente os **requeridos** para a entidade fato em análise.

#### ETL - Extrair, Transformar e Carregar:

ETL é o **conjunto de transações** de banco de dados usado para extrair informações de um banco de dados, transformá-los e carregá-los em um segundo banco de dados. As fontes de dados estão em aplicativos ou sistemas que são **externos** ao DW (fora da fronteira).

Lembre-se que a **intenção primária** dessas transações é **manter** dados lógicos (ou alterar o comportamento do sistema). As lógicas que envolvem os processos de **transformação** e **carga** de dados são secundárias à sua intenção primária.

Ou seja, **não** conte três *Processos Elementares* separados para cada uma das atividades (Ex: uma EE de Extração, uma EE de Transformação, e uma CE de Carregamento). Apenas o **conjunto** é auto-contido, uma vez que todas as três etapas são requeridas para completar a transação.

#### Arquivos de Interface Externa:

Durante o processo de ETL, caso haja a necessidade do processo acessar informações para fins de **referência** e **validação**, a origem destes dados deve ser analisada a fim de verificar se atende às regras de identificação de um **AIE**.

Note que os dados sendo diretamente extraídos, transformados e carregados pelo processo de ETL constituem **TD** da transação e **não** AIEs como Arquivos Referenciados da função.

Atente-se também ao fato de que em muitas situações estas tabelas de origem **negocial** externas à aplicação de DW são **copiadas**, fisicamente (sem nenhuma lógica especial de processamento) para dentro do próprio DW, a fim de atender a requisitos **não-funcionais** da aplicação (Ex: desempenho; arquitetura; etc) e, portanto, devem ser reconhecidos normalmente como **AIEs** da aplicação quando utilizados para fins de **referência** e **validação**.

### **6.11. Workflow**

---

Para correta medição de sistemas de workflow, é essencial avaliar e definir corretamente a sua **fronteira**, seus grupos de **usuários** e a visão de suas **transações** (a partir de uma perspectiva do negócio).

Um sistema de workflow é a automação de um **processo de negócio**, no todo ou em parte, durante o qual documentos, informações ou tarefas são passadas de um participante para outro, para ação em conformidade com um conjunto de regras específicas.

Ferramentas cuja finalidade seja a de criar/manter aplicações de workflow de forma automatizada **não** devem ser enquadradas na mesma fronteira que o workflow em si. A maneira pela qual o fluxo de trabalho será construído (manualmente ou por meio de uma ferramenta de apoio) **não** deve interferir na definição da fronteira e escopo da contagem, uma vez que o workflow será o alvo de medição e não a ferramenta criada para automatizar a construção e manutenção destas aplicações (que pode ou não ser utilizada). Ou seja, deve-se avaliar as funcionalidades **disponibilizadas** ao usuário por meio do workflow e **não a forma** pela qual estas funções foram disponibilizadas.

No entanto, o maior desafio na medição deste tipo de aplicação está na identificação de suas **funções transacionais**. O erro mais comum cometido está relacionado à identificação da **menor** unidade de atividade com significado para o usuário, uma vez que as funcionalidades não são “quebradas” corretamente em diversos PEs até a sua menor atividade.

#### **6.11.1. Cenário I: Exemplo de Workflow**

---

##### Cenário:

*O processo de negócio “Solicitação de Viagem”, é composto por diversas tarefas que, por definição, devem acontecer em um ponto consistente no processo e com um conjunto definido de entradas e saídas, antes e depois da execução da atividade.*

*Este processo é composto pelas seguintes funcionalidades: Incluir Solicitação de Viagem, Editar Solicitação de Viagem, Concluir Solicitação de Viagem, Consultar Detalhes da Solicitação de Viagem e Pesquisar a Solicitação de Viagem. Cada uma destas funcionalidades é completa em si mesma e deixa a aplicação em um estado consistente. Por exemplo, o usuário pode incluir uma solicitação de viagem e salvar seus dados como um rascunho e encaminhar adiante.*

O processo de negócio “Solicitação de Viagem”, de forma genérica, incluindo todas as atividades relacionadas à solicitação, **não** é a menor unidade de atividade com significado para o usuário final, apesar de ser completa e deixar o sistema em um estado consistente.

A atividade de “Editar Solicitação de Viagem” parte da tarefa “Solicitação de Viagem” também é **completa** e deixa o sistema em estado **consistente**. Uma vez

concluída, o usuário pode consultar em outro momento posterior o trabalho que fez, pode alterar novamente os dados que informou e passar o documento adiante.

Não existe atividade **menor** do que esta que seja completa. Preencher o centro de custo, por exemplo, dissociado do preenchimento dos demais dados não é completo de uma perspectiva do negócio. Assim, a inclusão da solicitação de viagem é um processo elementar, bem como:

- Solicitação de viagem – Incluir
- Solicitação de viagem – Editar
- Solicitação de viagem – Concluir
- Solicitação de viagem – Consultar detalhes
- Solicitação de viagem – Pesquisar

Vale ressaltar que, caso fosse utilizada uma ferramenta de automação, a análise feita sobre o cenário apresentado continuaria a **mesma**. São as funcionalidades disponibilizadas ao usuário para que este atenda à sua necessidade do negócio que serão avaliadas, e não a forma pela qual estas foram geradas.

Este conceito também é aplicado sobre a avaliação da **unicidade** das funções. Portanto, caso haja outra pesquisa relacionada a um outro fluxo específico do workflow, esta deverá ser identificada como uma **nova** transação (respeitando as regras impostas pelo CPM), independente da ferramenta de apoio gerá-las de forma similar.

## **6.12. Views/Visões em Bancos de Dados**

Para correta medição de views em bancos de dados relacionais, é importante analisar as **causas** que justificaram a sua criação e a quais **objetivos** estas visam atender.

De modo geral, as visões são criadas para atender a requisitos técnicos do sistema (Ex: consolidar dados, facilitar a recuperação de informações, melhorar o desempenho da aplicação, etc) e, portanto, uma view em si **não** deve influenciar o **valor** da contagem de Pontos de Função. No entanto, esta pode indicar a existência de um ou mais **Arquivos Lógicos**, servindo como um **indício** no momento da identificação das funções de dado da aplicação.

### **6.12.1. Cenário I: Views/Visões úteis durante a identificação de Arquivos Lógicos**

Em diversos sistemas do FNDE, foram criadas views **públicas** (visíveis para todas as aplicações) com intuito de contornar uma **restrição de segurança/acesso** a bancos de

dados de aplicações distintas, onde sistemas externos não possuem acesso direto às informações mantidas na base de outra aplicação.

A partir de uma perspectiva do **negócio**, as views são transparentes, uma vez que são componentes puramente **técnicos**. Se a restrição de segurança não existisse, as aplicações não necessitariam das visões para **acessar** as informações desejadas e, portanto, poderiam existir apenas para atender a requisitos de performance, por exemplo.

Analisando o motivo pelo qual as views foram criadas, há fortes **indícios** de que, na contagem da aplicação externa que está acessando a view, deve ser contado pelo menos um **Arquivo de Interface Externa** para cada view utilizada. É necessário analisar as tabelas e campos que compõem a view e verificar quais **entidades do negócio** da aplicação estão sendo referenciadas pela visão, além de quais campos são efetivamente utilizados pelas aplicações que utilizam estas views.

Do ponto de vista da aplicação que mantém a view, a view não pode ser considerada um Arquivo Lógico, uma vez que não representa um requisito do usuário e não é mantida pela aplicação.

#### **6.12.2. Cenário II: Views/Visões não úteis durante a identificação de Arquivos Lógicos**

---

Em diversos sistemas do FNDE, foram criadas views **internas** com intuito de melhorar o **desempenho** da aplicação e **facilitar** a recuperação de dados pela equipe de implementação.

A partir de uma perspectiva do **negócio**, as views são transparentes, uma vez que são componentes puramente **técnicos**. Se os requisitos não-funcionais que justificaram a criação das visões não existissem, em nada afetaria a aplicação a partir de uma visão do **negócio**.

Analisando o motivo pelo qual as views foram criadas, **não** é possível concluir que possa haver **Arquivos de Interface Externa** na aplicação, já que estas são internas ao sistema.

Lembrando que **não** é aconselhável utilizar as views como insumo para a identificação de **Arquivos Lógicos Internos**, uma vez que estas são puramente técnicas e podem levar a erros de medição. Procure sempre utilizar os insumos mais próximos do mundo lógico da aplicação e evite, se possível, aqueles que estejam muito ligados a solução técnica do sistema.

Da mesma forma, neste cenário a view também não pode ser considerada um Arquivo Lógico, uma vez que não representa um requisito do usuário e não é mantida pela aplicação.

#### **6.13. Transações compartilhadas entre várias aplicações**

---

Para correta análise sobre funcionalidades sendo compartilhadas entre diversas aplicações, é importante verificar qual a **visão do usuário** sobre estas funções. Se o usuário reconhecer que está acessando uma transação **externa** à aplicação que estava utilizando, **apenas** a função mantida externamente seria reconhecida.

Porém existem casos em que há **funcionalidades replicadas** em diversos sistemas, sem haver uma reutilização das funções já disponíveis em outras aplicações. Apesar de não desejável, é um cenário comum e que acarreta na contagem de **mais de uma** função transacional.

A dica geral para auxiliar na distinção de tais cenários é avaliar se a funcionalidade passou por um **processo de desenvolvimento** de software ou não. Caso tenha passado, caracteriza uma **nova transação**, própria da aplicação em que está inserida. Observe que alterações na funcionalidade **central** não impactarão sobre esta, que exigirá novas manutenções para que se adéqüe aquela.

No entanto, o principal **cuidado** a ser tomado é com relação à construção de **novas** funcionalidades. A abordagem descrita acima é útil na verificação das transações em sistemas **legados**, cujo cuidado para evitar a replicação de funcionalidades não foi tomado. Porém no caso de novas funções, deve-se evitar a replicação de funcionalidades e as **transações corporativas** devem ser utilizadas sempre que possível. Caso esta orientação não seja seguida, o fornecedor deverá realizar as **correções** devidas como parte da garantia pelo serviço prestado.

#### **6.14. Consultas com múltiplas mídias**

---

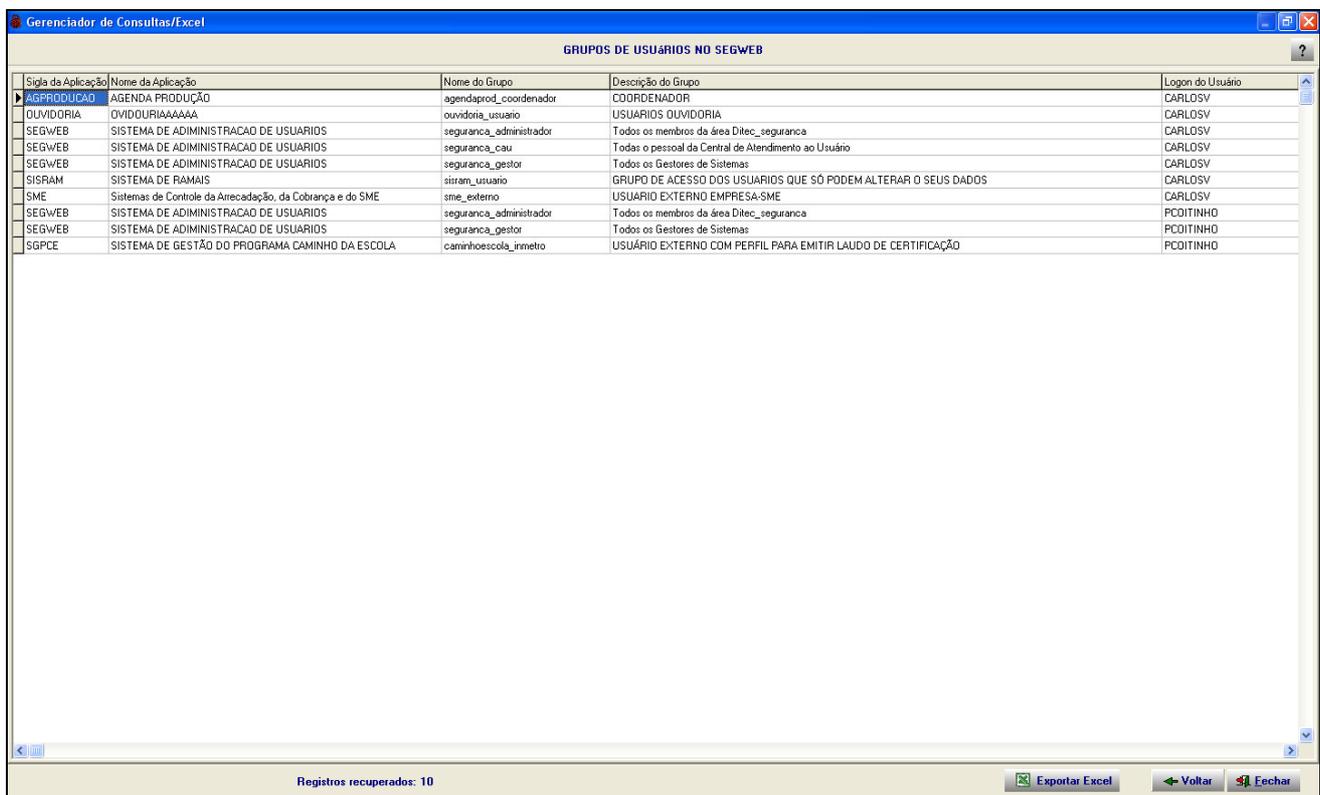
Basicamente a questão das múltiplas mídias consiste em um processo com a capacidade de gerar a sua **saída** ou processar a **entrada** de diversos meios (mídias) distintos e a divergência que surge é com relação à quantidade de Processos Elementares que deverão ser identificados (uma **única** transação ou um PE para **cada** mídia). O IFPUG reconhece duas abordagens possíveis para a questão: única instância e múltiplas instâncias para as funções que possuem várias mídias. A abordagem usada é decisão local de cada organização.

Para cobrir essa lacuna no manual do IFPUG, **serão contadas** todas as funções que deverão ser **desenvolvidas** para atender a uma mídia específica, desde que se trate de um requisito funcional especificado pelo usuário, enquanto que aquelas que forem disponibilizadas **automaticamente** por meio de componentes, **não** participarão da medição. A orientação é verificar quais funcionalidades devem ser testadas, com base em seus **casos de teste**. Esta orientação está em acordo com o descrito no próprio Manual de Práticas de Contagem:

"Outra indicação de que uma função de transação deve ser contada seria a cobertura de casos de testes. Um único grupo de casos de testes indicaria que um único processo elementar foi alterado."

## 6.14.1. Cenário I: Relatório com múltiplas mídias

Um relatório em forma de lista é gerado a partir de uma funcionalidade específica presente do sistema SegAdm. Além dos valores apresentados em tela, é possível exportar os resultados listados para uma planilha **Excel**, conforme ilustrado a seguir:



Sigla da Aplicação	Nome da Aplicação	Nome do Grupo	Descrição do Grupo	Logon do Usuário
AGPRODUÇÃO	AGENDA PRODUÇÃO	agendaprod_coordenador	COORDENADOR	CARLOSV
OUVIDORIA	OUVIDORIAAAAAA	ouvidoria_usuario	USUARIOS OUVIDORIA	CARLOSV
SEGWEB	SISTEMA DE ADMINISTRACAO DE USUARIOS	seguranca_administrador	Todos os membros da área Ditec_seguranca	CARLOSV
SEGWEB	SISTEMA DE ADMINISTRACAO DE USUARIOS	seguranca_cau	Todos o pessoal da Central de Atendimento ao Usuário	CARLOSV
SEGWEB	SISTEMA DE ADMINISTRACAO DE USUARIOS	seguranca_gestor	Todos os Gestores de Sistemas	CARLOSV
SISRAM	SISTEMA DE RAMAIS	sistem_usuario	GRUPO DE ACESSO DDS USUARIOS QUE SÓ PODEM ALTERAR O SEUS DADOS	CARLOSV
SME	Sistemas de Controle da Arecadação, da Cobrança e do SME	sme_externo	USUARIO EXTERNO EMPRESA-SME	CARLOSV
SEGWEB	SISTEMA DE ADMINISTRACAO DE USUARIOS	seguranca_administrador	Todos os membros da área Ditec_seguranca	PCOITINHO
SEGWEB	SISTEMA DE ADMINISTRACAO DE USUARIOS	seguranca_gestor	Todos os Gestores de Sistemas	PCOITINHO
SGPCE	SISTEMA DE GESTÃO DO PROGRAMA CAMINHO DA ESCOLA	caminhoescola_inmetro	USUÁRIO EXTERNO CDM PERFIL PARA EMITIR LAUDO DE CERTIFICAÇÃO	PCOITINHO

Registros recuperados: 10

Exportar Excel Voltar Fechar

SegAdm – Manual do Usuário

A funcionalidade responsável pela apresentação da lista em **tela** é facilmente identificada como um Processo Elementar, porém a habilidade de exportar o resultado para uma planilha Excel exige uma análise um pouco mais apurada.

Seguindo a orientação apresentada anteriormente, deve-se verificar se a capacidade de exportação foi disponibilizada por meio de um **componente** ou **framework** específico (onde apenas parâmetros de configuração são definidos para que a opção fique disponível) ou se esta funcionalidade passou por um **processo de desenvolvimento** de software.

A orientação é verificar os **casos de teste** utilizados e verificar a forma pela qual a funcionalidade foi tratada durante a sua criação. Havendo casos de teste específicos para a

habilidade de exportação, há um forte **indício** de que esta funcionalidade passou por um processo de desenvolvimento e não foi disponibilizada apenas por meio de uma configuração em um componente/framework da aplicação. Desta forma, a função de exportação seria reconhecida como um **PE** da aplicação.

### **6.15. Funções transacionais acessadas por diversos perfis de usuário**

Sistemas com vários perfis de usuário costumam ter transações (telas) compartilhadas por mais de um perfil. Normalmente cada perfil de usuário costuma representar um tipo de usuário no mundo do negócio tratado pelo sistema (exemplo em uma agência bancária: cliente, escriturário, caixa, tesoureiro, gerente). E cada tipo de usuário tem seus requisitos para o sistema; alguns requisitos são comuns a vários tipos de usuário e outros são específicos.

Cabe ao analista do sistema identificar os requisitos comuns a estes vários tipos de usuário e avaliar se é interessante tratá-los como um único requisito. Quando os requisitos são idênticos, esta deveria ser a abordagem seguida. Quando há algumas pequenas diferenças nestes requisitos comuns, mais análise é necessária. A decisão de tratar como requisitos distintos implicará na construção de mais artefatos no projeto, como por exemplo, diferentes casos de uso e telas distintas no protótipo. Basicamente esta é tipicamente uma decisão de modelagem de processos de negócio (e que foge ao escopo deste documento). Mas qualquer que seja a decisão tomada pelo analista do sistema, ela é dependente de aprovação (seja por parte do gestor do negócio, do analista de negócio, do usuário ou do patrocinador do projeto). Ou seja, esta decisão acaba se consolidando como a visão do usuário.

O manual do IFPUG, nas regras de identificação do processo elementar, diz que os requisitos funcionais do usuário devem ser decompostos nas menores unidades de atividade com significado para o usuário, que constituam uma transação completa, que sejam auto-contidas e deixem a aplicação num estado consistente.

E nas regras de unicidade do processo elementar, quando se comparam dois processos com o objetivo de identificar se são distintos ou não, deve-se verificar se possuem:

- o mesmo conjunto de tipos de dados e
- o mesmo conjunto de arquivos referenciados e
- o mesmo conjunto de lógicas de processamento.

Importante destacar que um processo pode ter pequenas variações nestes itens, porém isto não é suficiente para quebrar este processo em vários. A regra de unicidade não é para identificar novos processos, mas apenas para diferenciar processos previamente identificados (pelas regras de identificação).

Voltando então à questão da transação acessível por mais de um perfil de usuário: para se decidir se conta-se um ou mais processos; deve-se buscar no projeto o(s) requisito(s) que

deu(ram) origem a ela. Os requisitos aprovados é que servirão de base para esta decisão. Mas vale ressaltar que a decisão de modelar um ou vários requisitos deve representar da melhor forma a necessidade do usuário. Esta não deve ser uma decisão motivada apenas por requisitos não funcionais (ex: clareza na documentação, facilidade de leitura).

Para sistemas legados para os quais não esteja disponível a documentação dos requisitos, a diretriz sugerida é seguir o que o sistema oferece atualmente ao usuário: se uma tela é acessível a vários usuários e há a alguns campos ou lógica de processamento com pequenas variações, conta-se uma transação para todos os perfis.

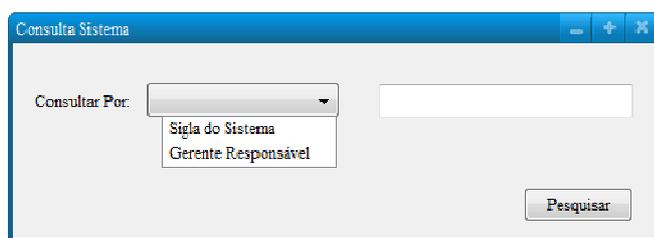
### 6.16. Consultas/Pesquisas com diversas opções de filtro

Para correta análise de funcionalidades de pesquisa, compostas por diversos campos e opções de filtro, é necessário verificar o seu **requisito funcional** e o que efetivamente foi solicitado pelo usuário da aplicação. O mais importante é conseguir separar corretamente quais opções adotadas durante a construção da funcionalidade foram tomadas com base em requisitos **não-funcionais**, evitando a contagem de diversas funções transacionais em situações em que um único Processo Elementar pode ser identificado.

#### 6.16.1. Cenário I: Consulta com um único Processo Elementar

O usuário solicita uma funcionalidade que será responsável pela **consulta** de sistemas a partir de sua **sigla** ou **gerente** responsável pela aplicação. O retorno da pesquisa, independente da opção de filtro selecionada, é sempre o **mesmo**.

A funcionalidade foi construída e disponibilizada ao usuário conforme ilustrado a seguir:



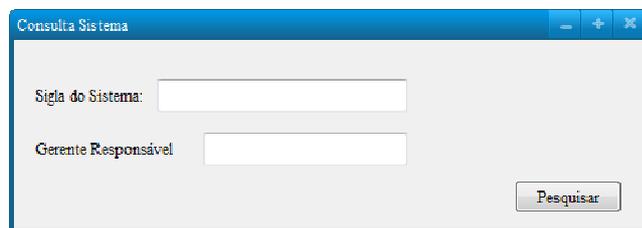
Observando apenas a funcionalidade já construída, pode-se chegar à interpretação **errônea** de que há **dois** PE distintos (Consultar por Sigla e Consultar por Gerente), uma vez que, a princípio, haveria a quebra de unicidade por conta dos tipos de dados distintos (caso estes não fossem replicados no retorno da pesquisa) e lógica de processamento distinta para cada caso. No entanto esta é uma interpretação **incorreta** e que pode acarretar em uma **grande distorção** na mensuração deste tipo de funcionalidade.

O que o usuário solicitou é que deve ser a informação utilizada com insumo para medição sendo realizada. O **requisito do usuário** é que determinará a quantidade de

Processos Elementares existentes e **não** a forma pela qual a solução foi construída. Neste exemplo, não há razões **funcionais** para que a consulta seja quebrada em duas transações distintas.

A dica geral é sempre fazer a analogia com uma tela de pesquisa em que todos os campos estivessem sempre disponíveis e avaliar qual o impacto no **negócio da aplicação**. Desta forma, decisões **não-funcionais** tomadas durante a implementação da funcionalidade ficam mais evidentes (Ex: Limitação de espaço em tela; Melhorar a usabilidade do sistema; etc).

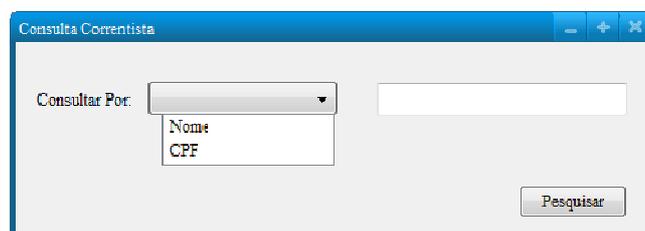
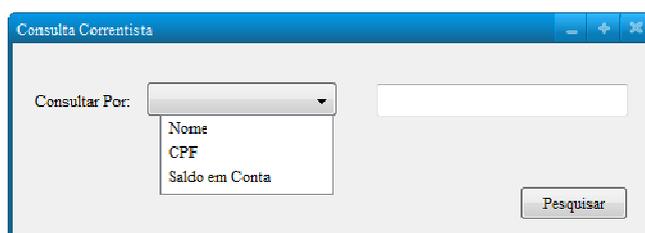
Note que para este exemplo, a tela poderia ter sido construída da seguinte forma, atendendo o requisito **funcional** da mesma maneira:



## 6.16.2. Cenário II: Consulta com vários Processos Elementares

O usuário solicita uma **consulta** de correntistas por **Nome** e **CPF** para os usuários com perfil de **Atendente**, e também por **Saldo em Conta**, caso o usuário possua perfil de **Gerente**. O retorno da pesquisa, independente da opção de filtro selecionada, é sempre o **mesmo**.

Desta forma, o atendente, ao acessar a transação de consulta, visualiza a tela ilustrada a seguir, enquanto que o gerente visualiza a tela subsequente.

A interpretação de ambas as transações quanto à contagem de um Processo Elementar para os campos **Nome** e **CPF** é a mesma já tratada no tópico 6.16.1 deste documento. Porém a interpretação quanto ao campo **Saldo em Conta** não é a mesma.

Note que existem usuários do negócio **distintos** com competências também distintas dentro da aplicação. Analisando sobre a ótica do **atendente**, existe uma única pesquisa com **duas** opções de filtro, enquanto que para o **gerente** há também uma única pesquisa, porém com **três** opções de filtro disponíveis. Ou seja, analisando a demanda como um todo, **dois** Processos Elementares distintos podem ser identificados: uma pesquisa por *Nome* e *CPF* e outra pesquisa por *Saldo em Conta*.

Para mais detalhes sobre a análise de transações compartilhadas por vários perfis distintos, vide tópico 6.15.

## **6.17. Trilha de Auditoria**

---

É muito comum em sistemas corporativos que haja o requisito de se rastrear as operações efetuadas pelo usuário. Normalmente isto é feito através da gravação de trilhas de auditoria que ficam disponíveis depois para consultas. A forma que a auditoria é implementada normalmente é definida pela área de segurança da informação da organização, que especifica que ações serão rastreadas, que dados serão armazenados, por quanto tempo, como serão consultados, etc.

### **6.17.1. Cenário I: Gravação da trilha de auditoria**

Normalmente a gravação da trilha de auditoria ocorre para todas (ou quase todas) as transações da aplicação. Não sendo, portanto, um **requisito específico e particular** destas transações, mas sim **requisitos gerais da aplicação**. Uma evidência disto, é que isto não estará presente nos casos de uso que especificam as transações, mas em uma parte geral da especificação de requisitos. Logo não há transação adicional a ser considerada pela gravação da trilha de auditoria; assim como também não haverá nestas transações a contagem de um arquivo referenciado para os dados de auditoria.

Casa haja um requisito específico e particular de gravar para determinado grupo lógico de dados informações de configurações passadas antes e após uma atualização, isso deve ser considerado na contagem. Isso se dá pela contagem de um novo tipo de registro nesse arquivo (afinal normalmente existe mais de um campo funcionalmente dependente dessa configuração anterior como a data em que o evento aconteceu, quem foi o responsável pela substituição da configuração anterior por uma nova ou mesmo pela exclusão de um registro).

A geração desse registro está associada ao seu fato gerador e com isso não deve ser contada uma função especificamente para esse fim, mas sim considerar a geração desse registro como um passo do processo responsável pelo fato gerador.

**6.17.2. Cenário II: Consulta à trilha de auditoria**

Caso seja responsabilidade da aplicação em análise disponibilizar consultas e relatórios aos dados de auditoria, estas transações deverão ser consideradas também funções desta aplicação, pois representam requisitos específicos da mesma.

Caso as consultas aos dados de auditoria sejam efetuadas por outra aplicação, nada relativo a estas consultas deve ser considerado na contagem da primeira aplicação.

Esta situação é bastante análoga ao exemplo de Telas de Help presentes no CPM.

**6.17.3. Cenário III: Dados da trilha de auditoria**

Na situação que em alguns grupos lógicos de dados da aplicação exigirem a manutenção do histórico de configurações anteriores; e houver necessidade de integrar esses dados em consultas que façam uma referência cruzada desses dados históricos independentemente do grupo lógico de dados em particular e da atual configuração desses dados; e os dados históricos são mantidos mesmo quando os dados atuais não foram mais relevantes para o negócio, cabe a contagem de um grupo lógico de dados especificamente para a sua manutenção.

Especificamente para as transações que mantem dados nesse arquivo, um novo AR será contabilizado ao avaliar a complexidade dessas funções.

## **7. Itens não Mensuráveis (INM) pela APF**

---

É importante destacar que a APF não mede todas as atividades relativas ao desenvolvimento e à manutenção de sistemas. Existem itens que não são mensuráveis pela técnica e **não** devem ser remunerados a parte, pois **já estão contemplados** no preço do PF acordado com o fornecedor.

No entanto, existem itens que não são medidos pela APF e não são responsabilidade ou não estão no escopo de trabalho acordado com o fornecedor. Neste caso, **deve-se** estabelecer um acordo para a remuneração do trabalho envolvido em tais atividades.

Algumas das atividades previstas foram estruturadas na forma de uma **tabela de itens não mensuráveis**, presente na própria planilha de contagem. Esta tabela contém os elementos/atividades que não são mensuráveis perante a técnica de Análise de Pontos de Função e um respectivo valor. Este valor é utilizado como uma medida de unidade de contrato, onde é atribuído um número equivalente ao PF. Para mais detalhes, vide o tópico 7.5 deste documento.

**ATENÇÃO:** Caso a manutenção que envolva requisitos não-funcionais seja realizada na mesma ordem de serviço de uma manutenção que altere requisitos de **negócio** na mesma função, **não** há remuneração adicional pela manutenção dos requisitos não funcionais, uma vez que se pagará pela alteração de toda a funcionalidade.

### **7.1. Manutenção Corretiva**

---

O IFPUG é muito claro com relação a este tipo de manutenção: como não há alteração de **requisito do usuário**, não há funcionalidade alterada e, portanto, **nada** a ser medido em PF.

O fato de a construção do sistema ter entregue algo diferente do especificado e solicitado pelo usuário é uma **falha** de responsabilidade do fornecedor. Este é o caso das demandas puramente corretivas. Na maioria das vezes, a correção de um defeito no software faz parte da **garantia** do serviço de desenvolvimento do fornecedor. Neste caso **não se mede** nada em PF e também não há nada a ser remunerado ao fornecedor.

Porém, para o caso em que o usuário constata que um **requisito** previamente aprovado está incorreto e a funcionalidade foi entregue exatamente conforme o especificado, não há defeito no software. Trata-se de uma alteração de funcionalidade que **deve** ser medida

pela APF como qualquer outra manutenção de um requisito de negócio e remunerada ao fornecedor.

Para tal, é **fundamental** que o requisito da solicitação a ser executada esteja **muito bem definido**. Caso contrário torna-se complicado avaliar o serviço em si, verificar se o mesmo apresentou de fato um defeito ou se o que foi entregue está de acordo com a especificação; se o problema ocorrido foi apenas um mau uso do sistema por parte do usuário; ou se houve mudança em relação ao escopo original.

Outra situação é a correção de um defeito realizada por um fornecedor que não foi o responsável pelo serviço que originou o defeito (Ex: Novo fornecedor assumindo sistema legado da contratante). Neste caso, seguindo estritamente a orientação do IFPUG, não há nada a ser **medido**. Porém há um trabalho a ser **remunerado** (uma vez que não se trata de garantia), cuja forma de remuneração deve ser combinada previamente.

No caso do FNDE, estas manutenções **corretivas** relacionadas a funcionalidades **fora da garantia** serão tratadas como **alteradas funcionalmente**, aplicando-se o mesmo deflator utilizado nestes casos. Estas entrarão automaticamente na **garantia** de correções no caso de futuros defeitos. Vale ressaltar que, caso o fornecedor alegue que outras funcionalidades tiveram de ser ajustadas para que o erro demandado pela contratante fosse corrigido, **todas** as funcionalidades listadas serão tratadas como **alteradas funcionalmente** e **todas** entrarão para **garantia**. Ou seja, mesmo que sejam identificados novos **erros** (não relacionados aos anteriores) nestas funcionalidades, estas **não** serão contadas e pagas novamente, enquanto durar o prazo de garantia definido em contrato.

No entanto, caso a manutenção corretiva **não** esteja associada a nenhuma funcionalidade **específica**, abrangendo características **gerais** da aplicação (Ex: infraestrutura), esta será tratada por meio da tabela de **Itens Não Mensuráveis**.

Também vale ressaltar que se a manutenção corretiva for realizada na mesma ordem de serviço de uma manutenção responsável por alterar requisitos de negócio na função, **não** há remuneração adicional pela manutenção corretiva, pois já estará sendo paga/medida a função por **completo**.

O FNDE adotará a seguinte política com relação à **garantia** do serviço para **correção de defeitos**: uma vez que uma função tenha sido alterada pelo fornecedor em atendimento a alguma manutenção evolutiva demandada pelo usuário, o fornecedor será responsável **integralmente** pela função, inclusive pelos possíveis defeitos pré-existentes. O controle do

FNDE observará as funções; aquelas que já tiverem sido alteradas na vigência do novo contrato estarão cobertas pela garantia, durante o prazo de garantia do contrato. O controle das funções que já foram alteradas ou criadas na vigência do novo contrato será através da planilha com o *baseline* de cada aplicação.

Caso seja necessária a construção de funcionalidades para a **correção de dados** da base de dados de um sistema, estas serão tratadas com base na tabela de **Itens Não Mensuráveis**, desde que os erros nos dados tenham sido ocasionados por defeitos pré-existent (fora da garantia do fornecedor), por uso indevido do sistema pelo usuário ou por alterações em outras aplicações que causaram impacto no sistema em questão. Porém, caso o defeito tenha ocorrido devido a erros em funcionalidades que se encontravam em garantia pelo fornecedor, as funções construídas para a correção dos dados **não** serão remuneradas, uma vez que estas fazem parte da garantia do serviço prestado originalmente.

Muitas vezes, a manifestação de um erro está longe de sua causa ou sequer é um erro propriamente dito, mas um mau uso do mesmo por parte do **usuário**. Como a investigação da causa do erro pode ser uma atividade demorada, caso haja o envolvimento do fornecedor na investigação de um erro de sistema e fique constado que a falha é do sistema pelo qual ele é responsável, o esforço gasto nesta atividade **não** deverá ser remunerado, pois já faz parte da garantia do serviço executado. No entanto, caso fique constatado que o erro é causado por mau uso do sistema ou que o erro está localizado em outro sistema, este serviço **deverá** ser remunerado como atendimento técnico, desde que **não** exija nenhum tipo de **ação** por parte do fornecedor para **correção** do problema.

Caso haja a necessidade do fornecedor corrigir o defeito, esta atividade será paga com base na tabela de **Itens Não Mensuráveis**. Neste caso, o esforço relacionado à atividade de investigação já estará incluído neste item e **não** será objeto de pagamento.

## 7.2. Manutenção Cosmética

---

Esta manutenção é caracterizada por mudanças na apresentação da função ao usuário ou também na forma como a entrada de dados é realizada. Exemplos:

- Mudança na ordem de tabulação de colunas;
- Estilo da interface, cores, fonte, títulos dos campos, botões;
- Mudança na máscara (formatação) do campo;
- Mudança na posição de campos na tela;
- O campo deve aceitar somente textos em maiúsculas ou minúsculas, não pode ter caracteres acentuados, etc;

Se a manutenção **não** envolver nenhum tipo de alteração funcional, de acordo com o IFPUG, trata-se de uma manutenção perfectiva, que **não** é objeto de medição para APF. Convém destacar que uma manutenção que visa apenas mudar a **ordenação** de uma consulta ou relatório, pode se enquadrar no critério de mudança de lógica **de processamento**, sendo uma manutenção passível de ser medida pela APF.

O FNDE adotará a seguinte política com relação às manutenções **cosméticas**: todas as **funções transacionais** afetadas pela manutenção serão identificadas e para cada uma destas funcionalidades será aplicado um **deflator** para obtenção do valor a ser pago, em equivalência o preço do PF. Ou seja, o valor irá variar com base na **complexidade funcional** do Processo Elementar (Vide tópico 7.5.1).

No entanto, caso a manutenção cosmética **não** esteja associada a nenhuma funcionalidade **específica**, abrangendo a aplicação como um todo, esta será tratada como mudança em características gerais da aplicação, conforme exposto no tópico 7.5.2 (Ex: Alteração de layout em folha de estilo do sistema; Alteração da disposição/texto dos menus de navegação do sistema; etc).

### **7.3. Manutenção Técnica (ou Adaptativa)**

---

Esta manutenção é caracterizada por alterações no software para atender a algum novo requisito **técnico** como mudanças de: sistema operacional, banco de dados, softwares de comunicação ou na infra-estrutura computacional de maneira geral. Pode ser também uma manutenção para alterar a forma de implementar uma transação com o intuito de melhorar portabilidade, manutenibilidade, performance, etc. Não há alteração de requisito de **negócio** envolvida. Portanto, este tipo de manutenção **não** é medido pela APF.

O FNDE adotará a seguinte política com relação às manutenções **adaptativas**: todas as **funções** afetadas pela manutenção serão identificadas e para cada uma destas funcionalidades será aplicado um **deflator** para obtenção do valor a ser pago, em equivalência ao preço do PF. Ou seja, o valor irá variar com base na **complexidade funcional** da funcionalidade (Vide tópico 7.5.1).

No entanto, caso a manutenção adaptativa **não** esteja associada a nenhuma funcionalidade **específica**, abrangendo a aplicação como um todo, esta será tratada como

uma mudança em características gerais da aplicação, conforme exposto no tópico 7.5.2, salvo nas situações em que uma aplicação esteja sendo construída por **completo** com as mesmas funcionalidades, alterando apenas a tecnologia utilizada. Neste caso o novo sistema será tratado normalmente como um **projeto de desenvolvimento**, sem aplicação de nenhum deflator.

Porém, se a manutenção técnica tiver o **mesmo escopo** e for realizada na mesma **solicitação** de uma manutenção que altere requisitos de **negócio** da função, **não** há remuneração adicional pela manutenção técnica, pois a função completa já estará sendo mensurada.

#### **7.4. Dados de código**

---

Conforme estabelecido pelo IFPUG, dados de código **não** devem ser reconhecidos como Arquivos Lógicos durante a medição, assim como as transações que trabalhem **exclusivamente** com este tipo de dado não devem participar da medição.

Com base neste entendimento, em um **projeto de desenvolvimento**, implementar tabelas que representam dados de código, bem como construir transações para operar sobre estes dados **não** deverão afetar a contagem de pontos de função e muito menos serem remunerados separadamente. O preço por PF estabelecido no contrato **deve** levar em considerações os requisitos técnicos e de qualidade que o cliente espera do serviço. O mesmo vale para os **projetos de melhoria** que, em função da manutenção, gere/altere dados de código sobre as funcionalidades alteradas.

No entanto, já as demandas que tratem **apenas** de dados de código serão remuneradas com base na tabela de **Itens Não Mensuráveis**. Ou seja, caso seja solicitada apenas a criação de uma tela de cadastro de dados código, por exemplo, esta demanda será enquadrada na tabela de Itens Não Mensuráveis, ao contrário de uma demanda que solicite a criação de uma **funcionalidade** e, além disso, a criação da tela de manutenção de seus dados de código. Neste caso não haverá o pagamento da funcionalidade e do respectivo Item Não Mensurável.

#### **7.5. Tabelas de Itens Não Mensuráveis**

---

Conforme preconizado pela técnica de Análise de Pontos de Função, nem todos os tipos de solicitações podem ser medidas em PF, uma vez que esta tem o foco única e

exclusivamente sobre os **requisitos funcionais** da aplicação, sem analisar “**como**” uma aplicação é construída, levando em consideração apenas “**o que**” é disponibilizado pela mesma.

No entanto, ao contrário do que muitos dizem, o **usuário** pode sim solicitar **requisitos não-funcionais** e inclusive o faz constantemente. Como exemplo, temos a abertura de uma demanda para alterar a interface gráfica da aplicação (estrutura do layout; cores; imagens; etc). Este tipo de solicitação não interfere no **negócio** da aplicação, porém foi demandada pelo usuário e deverá ser atendida, gerando **esforço** por parte do **fornecedor**.

A fim de enquadrar estes tipos de demanda, será adotada uma tabela de **Itens Não Mensuráveis** que estará relacionada a requisitos **não-funcionais** associados **diretamente** a **funções** da aplicação (requisitos **funcionais**). Em relação às demandas que afetem apenas as **características gerais** de uma aplicação, sua remuneração será realizada diretamente em horas/atividade, conforme exposto no tópico 7.5.2.

## 7.5.1. INM associados a funcionalidades

São aqueles que incidem diretamente sobre uma **função** da aplicação, porém sem afetar requisitos **funcionais**. Para estes itens será aplicado um **deflato**r sobre a complexidade da função afetada. Ou seja, a funcionalidade deverá ser contada normalmente e relacionada ao INM específico para que, com base em sua contribuição, chegue-se ao **valor de equivalência** em PF.

INM	Deflato	Descrição
Code Table	20%	<p>Contempla a necessidade de criação, alteração e exclusão de tabelas CODE DATA e respectivas funcionalidades na aplicação, ou seja, mantida pelo usuário.</p> <p>Serão considerados 20% do valor das funções de dados e/ou transacionais, caso fossem mensuráveis no tamanho funcional do projeto.</p> <p>Vale ressaltar que, apesar de serem considerados como itens não mensuráveis, tais tabelas não são ALI's ou AIE's da aplicação, e portanto, não podem ser consideradas Arquivos Referenciados nos processos elementares contados para a aplicação.</p>

Manutenção Cosmética	10%	<p>Contempla as manutenções associadas exclusivamente à alterações de interface/layout de forma geral. Para mais detalhes, vide tópico 7.2.</p> <p>No entanto, apesar de serem manutenções cosméticas, alterações referentes a ajustes nos menus de navegação do sistema e em sua folha de estilo não serão enquadrados neste item. Estes serão tratados com base nas regras definidas no tópico 7.5.2.</p>
Manutenção Corretiva fora da Garantia	60%	<p>Contempla as atividades de correção de defeitos em uma funcionalidade entregue, a fim de garantir que esta funcione de acordo com a sua respectiva especificação.</p> <p>Só serão enquadradas neste item as manutenções corretivas que estiverem fora de garantia. Para mais detalhes, vide tópico 7.1.</p>
Manutenção Adaptativa em Requisitos Não-Funcionais	50%	<p>Contempla a adequação do sistema às mudanças de ambiente operacional, compreendendo hardware e software básico, mudanças de versão, linguagem e SGBD, que não impliquem em inserção, alteração ou exclusão de funcionalidades. (Vide tópico 7.3).</p>
Manutenção Adaptativa para re-desenvolvimento em outra plataforma	100%	<p>Contempla a adequação do sistema às mudanças de ambiente operacional, de modo que seja necessário re-construir a aplicação por completo. Para mais detalhes, vide tópico 7.3.</p>
Manutenção de Documentação de Sistemas Legados	20%	<p>Contempla as demandas de criação ou atualização de documentação, solicitadas de forma isolada. Ou seja, a solicitação deve estar desassociada de manutenções funcionais, onde a atividade de criação/atualização de documentação já está compreendida.</p>
Função em Garantia	0%	<p>Contempla as demandas em garantia que apresentem erros em sua execução, não atendendo corretamente ao seu respectivo requisito funcional. Para mais detalhes, vide tópico 7.1.</p>

### 7.5.2. INM associados a características gerais da aplicação

São aqueles que **não** incidem diretamente sobre uma **função** da aplicação e sim sobre o sistema como um todo. Para estes itens a remuneração será realizada com base no esforço depreendido para a execução do serviço (horas/atividade), desde que sejam entregues os produtos solicitados dentro do padrão de qualidade definido pelo FNDE.

## **8. Escopo do Serviço em PF**

---

### **8.1. Atividades Inclusas no Preço do PF**

---

Basicamente todas as atividades que possuem relação direta com a entrega dos requisitos funcionais solicitados pelo usuário: levantamento e especificação de requisitos, análise, projeto, modelagem, gerência do projeto, codificação, testes, apoio à homologação do usuário, implantação e transferência de conhecimento do serviço executado. Sendo que vários artefatos podem ser produzidos nestas atividades, tais como: código fonte, diagramas, modelos, casos de uso, manuais, planos, atas, etc.

### **8.2. Atividades NÃO Inclusas no Preço do PF**

---

Por complemento ao item anterior, quaisquer atividades não diretamente relacionadas ao desenvolvimento de software não estão no escopo do serviço contratado por PF. Exemplos: treinamento de usuários (dissociado do projeto de desenvolvimento / melhoria), acompanhamento do sistema em produção, administração do banco de dados, atendimento de dúvidas ou reclamações de usuários, atividades de suporte a infra-estrutura tecnológica, etc.

A remuneração destas atividades, quando necessárias, será realizada conforme exposto no conforme exposto no tópico 7.5.2.

## 9. Referências Bibliográficas

---

- i. Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software - Livro que ensina todo o processo de contagem de pontos de função, de forma aderente à última versão do Manual do IFPUG. Vide [www.fattocs.com.br/livro.asp](http://www.fattocs.com.br/livro.asp);
- ii. Manual de Práticas de Contagem de Pontos de Função, versão 4.3 – [www.ifpug.org](http://www.ifpug.org);
- iii. IFPUG White Paper - Function Points & Counting Enterprise Data Warehouses
- iv. IFPUG White Paper - Function Points & Counting Middleware Software Applications
- v. PMBoK, 4ª Edição – [www.pmi.org](http://www.pmi.org);
- vi. Fóruns de Discussão sobre APF
  - a. IFPUG - <http://www.ifpug.org/webforum/discus/discus.cgi>
  - b. BFPUG - <http://br.groups.yahoo.com/group/forum-bfpug/>
  - c. Livro APF - <http://br.groups.yahoo.com/group/livro-apf/>
- vii. Glossário Interativo de APF
  - a. Contém todos os termos utilizados no contexto da APF, com definição completa e exemplos. Vide [www.fattocs.com.br/glossario.asp](http://www.fattocs.com.br/glossario.asp)